

Mika Mähönen

NETSURVEY-KYSELYIDEN MUODOSTUS- JA ANALYSOINTI-  
OHJELMA

Tietotekniikan koulutusohjelma

Ohjelmistotekniikka

2009

## NETSURVEY-KYSELYIDEN MUODOSTUS- JA ANALYSOINTIOHJELMA

Mähönen, Mika

Satakunnan ammattikorkeakoulu

Tekniikka ja merenkulku Rauma

Tietotekniikan koulutusohjelma

Huuhtikuu 2009

Yritys: Satakunnan ammattikorkeakoulu

Ohjaaja: FM. Ritva Kompuinen

UDK: 004.42

Sivumäärä: 58

Asiasanat: Ajax-ohjelmointi, JavaScript, PHP, SQL

Tämän opinnäytetyön tarkoituksena oli luoda ohjelma, jota voidaan käyttää Web-kyselyiden luomiseen ja analysoimiseen dynaamisesti. Alkuperäinen idea tämän työn toteuttamiseen tuli Satakunnan ammattikorkeakoululta.

Opinnäytetyön toteuttaminen muodostui kolmesta työvaiheesta, jotka olivat suunnittelu, ohjelmointi ja vikojen korjaaminen. Suunnitteluvaiheessa kaikki ohjelman ominaisuudet suunniteltiin huolella ohjelmointivaihetta varten. Tässä vaiheessa myös ohjelman toteutukselle asetettiin tavoitteet, jotka olivat joustavuus, turvallisuus ja monikielisyys. Sen jälkeen kun ohjelman suunnittelu oli tehty, valittiin projektia varten tietokantamoottori sekä sopivat ohjelmointikielet tavoitteiden toteuttamiseksi. Ohjelmointivaiheessa kaikki Web-lomakkeet toteutettiin hyödyntäen Php:tä, JavaScriptiä ja Ajaxia. Kyseinen vaihe oli myös työn pitkäkestoisin prosessi, koska suunnitteluvaiheessa ohjelmaan oli valittu paljon toteutettavia ominaisuuksia. Viimeisessä työvaiheessa testattiin luotu ohjelmakoodi ja korjattiin siitä löytyneet viat.

Tuloksena tästä opinnäytetyöstä saatiin ohjelma, jota voidaan käyttää esimerkiksi työpaikkaviihtyvyys- ja koulutustarvekyselyn toteuttamiseen. Opinnäytetyö on myös herättänyt yleistä kiinnostusta erilaisissa organisaatioissa sen joustavuuden ja monipuolisen sovellettavuuden ansiosta eri käyttötarpeisiin.

# NETSURVEY, COMPUTER-ASSISTED SURVEY FORMING AND ANALYZING TOOL

Mähönen, Mika

Satakunta University of Applied Sciences

Technology and Maritime Management Rauma

Degree Programme in Information Technology

April 2009

Commissioned by Satakunta University of Applied Sciences

Tutor: Ritva Kompuinen, MSc

UDC: 004.42

Number of pages: 58

Keywords: Ajax programming, JavaScript, PHP, SQL

The purpose of this thesis was to create a dynamic program which could be used for forming surveys, and receiving and analyzing responses. The original idea of doing this project was received from Satakunta University of Applied Sciences.

The creation of this thesis comprised three different working phases: designing, programming and testing. In the design phase all the features of the program were carefully planned for the programming phase. Also major development targets were set for creating the program. They were flexibility, security, and multilingual support of Web forms. After obtaining the list of created features to be implemented, a suitable database engine and programming languages were selected for the project. In the programming phase all the Web forms were created using Php, JavaScript and Ajax programming. The programming phase was also the most time consuming-phase during the study because of the large number of designed and implemented features. The last phase consisted of testing the created program code and correcting flaws.

The result of this project was a survey forming tool which could be applied to many different scientific researches and surveys. This program could be used for example for the following surveys: analysis of job satisfaction, evaluation of education needs within a company etc. This project has also aroused several organizations' interest due to the suitability of the tool for different kinds of purposes.

# SISÄLLYS

LYHENTEET.....	6
1 JOHDANTO.....	7
2 KÄSITTEET.....	8
2.1 Html, Xhtml ja Get- ja Post-metodi.....	8
2.2 Php.....	10
2.3 InnoDB.....	11
2.4 Ajax ja JavaScript.....	13
2.5 Serialisaatio.....	14
2.6 Kaavioiden piirtokirjastot.....	14
2.7 Captcha.....	15
2.8 Mysql-injektio.....	15
2.9 Md5-tiivisteet, Saltaus-menetelmä ja Rijndael.....	17
3 OHJELMAN TOTEUTTAMINEN.....	18
3.1 Ohjelman suunnittelu.....	18
3.1.1 Ohjelman kehitysympäristön valitseminen.....	18
3.1.2 Toteutettavien ominaisuuksien suunnittelu.....	18
3.1.3 Sivuston rakenteen suunnitteleminen.....	21
3.1.4 Tietokanta.....	26
3.2 Toteutus.....	28
3.2.1 Kyselyhallintaan kirjautuminen.....	28
3.2.2 Omat tiedot.....	32
3.2.3 Käyttäjähallinta.....	38
3.2.4 Lisää kysely.....	39
3.2.5 Etsi kyselyä.....	40
3.2.6 Muokkaa kyselyä.....	41
3.2.7 Muokkaa kyselyn käyttöoikeuksia.....	43
3.2.8 Poista kysely.....	44
3.2.9 Muokkaa kyselyn kysymyksiä.....	45
3.2.10 Vastaajaryhmät.....	47
3.2.11 Kyselyyn vastaaminen.....	48
3.2.12 Tuloksien tarkastelu.....	49
3.3 Virheiden etsintä ja korjaus.....	51
4 YHTEENVETO.....	53
LÄHTEET.....	54

LIITTEET.....	56
---------------	----

## LYHENTEET

AES	Advanced Encryption Standard
AJAX	Asynchronous JavaScript and XML
ASP	Active Server Pages
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
http	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HTML	Hyper Text Markup Language
JSP	JavaServer Pages
MD5	Message-Digest
MITM	Man-in-the-middle
PHP	Hypertext Preprocessor
SQL	Structured Query Language
WWW	World Wide Web
XML	Extensible Markup Language
XHTML	Extensible HyperText Markup Language

# 1 JOHDANTO

Opinnäytetyön tavoitteena oli luoda Satakunnan ammattikorkeakoululle helppokäyttöinen ohjelma, jolla voidaan muodostaa kyselylomakkeilta sekä tarkastella kyselylomakkeista saatuja vastauksia. Varsinainen idea työn toteuttamiseksi tuli Satakunnan ammattikorkeakoululta.

Erityistavoitteena ohjelman kehittämisessä oli, että sitä pystyttäisiin käyttämään monikansallisessa ympäristössä. Muita merkittäviä tavoitteita olivat tietoturvallisuus ja ohjelman tuki yleisimmille kysymystypeille, joita esitetään vastaajille suoritettavan kyselyn aikana. Ohjelma suunniteltiin niin, että sitä voidaan käyttää tulevaisuudessa esimerkiksi henkilöstön viihtyvyyskyselyn laatimiseen sekä saatujen tulosten arviointiin. Työn tekemiseen päädyttiin, koska vastaavaa ohjelmaa markkinoilla ei ole saatavilla ilmaiseksi tällä hetkellä.

## 2 KÄSITTEET

### 2.1 Html, Xhtml ja Get- ja Post-metodi

Html eli Hyper Text Markup Language. Sitä käytetään yleisesti sivustojen kuvaamiseen niin sanottujen Html-tagien avulla. Itse Html:stä internetissä on käytössä monia eri versioita, joista työhön on käytetty Xhtml 1.0 STRICT -standardia. Xhtml eli EXtensible HyperText Markup Language on tällä hetkellä W3Schoolsin antama suositus käytettävästä standardista. Yksinkertaisimmillaan Html- tai Xhtml-sivun kuvaaminen näyttää seuraavalta:

```
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```



Kuva 1. Esitellystä Html-koodista muodostettava sivu.

Html:n ja Xhtml:n keskeisin ongelma on staattisuus, jolla tarkoitetaan sitä, ettei käyttäjä voi itse vaikuttaa palvelimelta saatuun sivuun. Tämä ongelma pystytään kuitenkin ratkaisemaan käyttämällä dynaamista skriptikieltä mm. Asp, Jsp tai Php. (W3Schools 2009.)

Vuorovaikutteisesta Html:stä puhutaan, kun käyttäjä voi antaa syötteitä esimerkiksi lomakkeelta. Tällöin käyttäjän antama syöte lähetetään palvelimelle käyttäen hyväksi

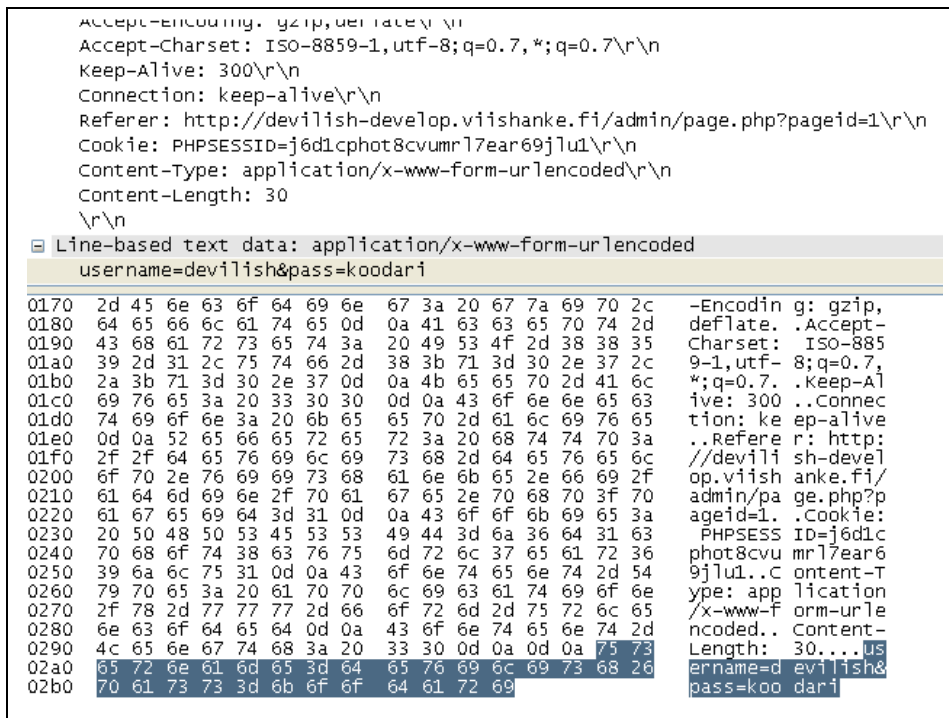


Head-, Get- tai Post-metodia, jolloin palvelimen ohjelmisto kykenee tunnistamaan syötteen sille tulevasta syötevirrasta sekä käsittelemään syötteen sisällön. Näkyvin ja suurin ero Post- ja Get-metodien välillä onkin tapa, jolla syötevirta ohjataan palvelimelle. Get-metodilla tiedot viedään hyödyntäen osoiteriviä, jolloin syötevirta luetaan suoraan absoluuttisesta osoitteesta. Palvelimelle vietävät muuttujat löytyvät ?-merkin jälkeen (kuva 2) ja mihin ne ovat erotettu toisistaan käyttäen hyväksi &-merkkiä.



Kuva 2. Get-metodin syötevirta osoiterivillä.

Get-metodin ongelmana on kuitenkin lähetettävän tiedon paljastuminen käyttäjälle jo osoitteesta, mikä puolestaan voi johtaa salasanojen sekä tunnuksien kohdalla tietoturvaongelmiin. Kyseistä ongelmaa voidaan kiertää käyttämällä Get-metodin sijasta Post-metodia paikoissa, joissa lähetettävää syötevirtaa ei haluta paljastaa käyttäjälle suoraan osana osoitetta. Post-metodissa muuttujien sisältö lähetään Http-protokollan kehykseen sijoitettuna osoitteen sijasta, jolloin tiedot eivät ole myöskään suoraan käyttäjän nähtävillä osoiterivillä.



Kuva 3. Http-protokollan paketti.

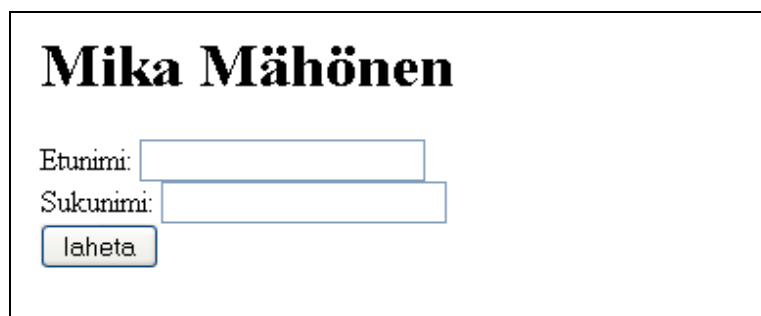
Näiden edellä mainittujen metodien lisäksi on olemassa Head-metodi, jossa syötevirta ohjataan Get-metodin tavoin palvelimelle. Tätä metodia voidaan käyttää, kun palvelimelta ei odoteta vastausta lähetettyyn syötteeseen. Tämä toimintamalli myös erottaa Head-metodin muista edellä mainituista metodeista. (Leponiemi 2002.)

## 2.2 Php

Php:tä voidaan pitää sulautettuna palvelin skriptikielenä, jonka kielioppi muistuttaa hyvin paljon C:tä, Javaa sekä Perliä. Itse Php tulee sanoista HyperText Preprocessor, ja sitä käytetään dynaamisten WWW-sovelluksien luomiseen. Php:n avulla toteutettuja sivustoja esimerkiksi ovat keskustelupalstat, Wikipedia ja eräiden pankkien sivut. Tällaisilla sivustoilla käyttäjän syöttämä tieto vaikuttaa luotavan Html-sivun sisältöön. Ilman dynaamista skriptikieltä jokainen käyttäjälle näytettävä sivu tulisi luoda staattisesti erikseen. (Tizag 2008.)

Seuraavassa on hyvin yksinkertainen esimerkki Php:n käytöstä Html-lomakkeella.

```
<html>
<body>
<?php
if(isset($_POST["etunimi"]) && isset($_POST["sukunimi"])) {
    ?><h1><?php echo $_POST["etunimi"].' '.$_POST["sukunimi"]; ?></h1><?php
}
?>
<form name="input" action="sivu.php" method="post">
Etunimi: <input type="text" name="etunimi"><br />
Sukunimi: <input type="text" name="sukunimi"><br />
<input type="submit" value="laheta">
</form>
</body>
</html>
```



**Mika Mähönen**

Etunimi:

Sukunimi:

Kuva 4. Tuloksena saatu dynaaminen Web-sivu.

## 2.3 InnoDB

InnoDB on tietokantaohjelmisto, johon NetSurvey-ohjelman sisältämät tiedot tallennetaan. Tarkemmin InnoDB on Mysql:n päälle rakennettu transaktionaalinen tallennusmoottori. Transaktionaalisella tallennuksella tarkoitetaan tässä yhteydessä tiedon eheyden säilyttämistä tietojenkäsittelyn yhteydessä. Käytännön esimerkki tiedon eheydestä saadaan, kun ohjelmasta poistetaan kysely. Tällöin ohjelmiston tulee myös huolehtia kyselyyn liittyneiden kysymyksien sekä niihin saatujen vastauksien poistamisesta. Keskeisin InnoDB:n ero juuri muihin Mysql-tallennusmoottoreihin on transaktionaalisuudessa. Esimerkiksi MyISAM-tallennusmoottori on relationaalinen tietokantamoottori, ja näin ollen se ei huolehdi tietojen eheydestä itse. Relationaalisen ja transaktionaalisen tallennusmoottorin ero on tavassa, jolla tietojen eheys säilytetään. Relationaalisessa mallissa viiteavainten päivityksistä tytärtauluihin ei huolehdita osana tallennusmoottoria, jolloin myös tiedon eheys voidaan helpommin menettää. Transaktionaalisessa mallissa huolehditaan puolestaan tiedon eheydestä suorittamalla toimenpide myös olemassa oleville viiteavaimille. Tämä on yksi suurimmista syistä miksi transaktionaalisen tietokannan käyttäminen on mielekkäämpää ohjelmoijan kannalta kuin relationaalisen. Tietyissä tilanteissa relationaalinen tietokantamalli on kuitenkin huomattavasti joustavampi ratkaisu. Opinnäytetyön kannalta ei ollut kuitenkaan perusteltua käyttää relationaalista rakennetta sen lisäämien ohjelmakoodirivien takia. Transaktionaalisiin tietokantoihin liittyvissä teksteissä viiteyhteyksien päivittämisestä käytetään usein myös termiä vyöryttäminen. (InnoDB 2008; Kekkonen 2009; Mysql 2008.)

Vyöryttäminen käsittää käytännössä kolme erillistä toimintamallia tietokannassa esitetyille Päivitä- ja Poista-toiminnoille.

CASCADE (Viiteavainten päivitys pääavaimen mukaiseksi):

Päätaulun avainta muutettaessa kohdistetaan muutokset myös tytärtaulujen viiteavaimiin. Ohjelmassa Cascade-toimintoa käytetään, kun surveyid:tä muokataan surveys-taulussa. Tämä johtaa viiteavainten päivittämiseen myös tytärtauluissa questions ja rights. Poistettaessa pääavain surveys-taulusta poistetaan myös tytärtaulujen viiteavain tietueet.

SET NULL (Viiteavaimen hävittäminen):

Päätaulun avainta muutettaessa tai poistettaessa asetetaan tytärtaulun viiteavaimen kenttä tyhjäksi, joten näin tietueiden välinen yhteys kadotetaan. Tämä mahdollistaa eräänlaisen on- tai ei ole -mallisen transaktion toteuttamisen tietokannan taulujen välillä. Ohjelmasta tällainen transaktion löytyy käytöstä, kun vastaajaryhmä poistetaan. Vastaajaryhmällä on transaktio tauluun kyselyt, jossa vastaajaryhmän poiston yhteydessä asetetaan tyhjäksi kyselyryhmä niiden kyselyiden osalta, joissa käytettiin poistettua vastaajaryhmää. Päivitä-toiminnossa SET NULL toimintoa ei ole käytetty ohjelmassa.

RESTRICT tai NO ACTION (Toimenpiteen estäminen):

Päätaulun avaimen päivitys tai poisto estetään. Tätä toimintoa ei käytetty missään transaktiossa luodussa tietokannassa. (Kekkonen 2009; Mysql 2008.)

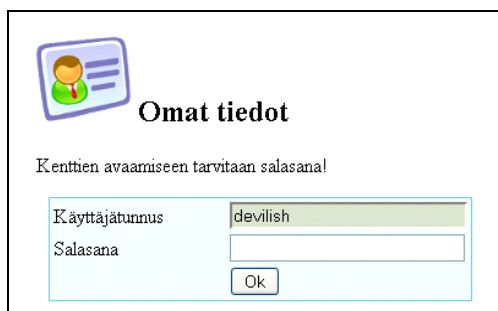
## 2.4 Ajax ja JavaScript

JavaScript on yleisesti selaimissa käytetty skriptikieli, jota käytetään selaimien ohjelmointiin vrt. Php, jota käytetään Web-palvelimen ohjelmointiin. Ajax eli Asynchronous JavaScript and XML on JavaScriptiin perustuva laajennus, joka täydentää JavaScriptin toimintoja. Internetissä on saatavilla useita erilaisia Ajax-laajennoksia, joista työssä päädyttiin käyttämään Prototype-nimistä kirjastoa. Se on kaikkien vapaasti haettavissa ilmaiseksi. Ajaxilla voidaan toteuttaa selaimiin älykkäitä toimintoja, kuten tietojen automaattista käsittelyä. Ajaxin keskeisin ongelma on kuitenkin tietoturva, koska sopivilla työkaluilla muuttujien arvoja voidaan muuttaa tahallisesti virheellisiksi. Edellä mainitun ongelman takia tarkistukset on tehtävä niin selaimen kuin palvelimenkin päässä, jotta tiedon sopivuudesta voidaan varmistua. (Prototypejs 2009.)

Opinnäytetyössä Ajaxin käytöstä on hyvä esimerkki Omat tiedot –lomakkeen toiminto, jossa käyttäjää pyydetään syöttämään salasana. Salasanan antamisen jälkeen se lähetetään automaattisesti oikealle sivulle käsiteltäväksi.



Kuva 5. Lomake, jossa tietojen lähetys hoidetaan käyttäen Ajaxia.



Kuva 6. Sama lomake ilman Ajaxia.

## 2.5 Serialisaatio

Serialisaatio tarkoittaa suomeksi objektia, joka sisältää joukon objekteja. Esimerkiksi monivalintakysymys sisältää ennalta tuntemattoman määrän vastauksia, jotka käyttäjä voi valita sisällyttäväksi yhteen vastaukseen. Tällaisessa tapauksessa on edullista hyödyntää serialisaatiota esittämään saatuja vastauksia yhtenä kokonaisena vastauksena. Tämä mahdollistaa esimerkiksi monivalintakysymyksestä saadun vastauksen pakkaamisen yhdeksi ainoaksi vastaukseksi. (Gervasio 2006.)

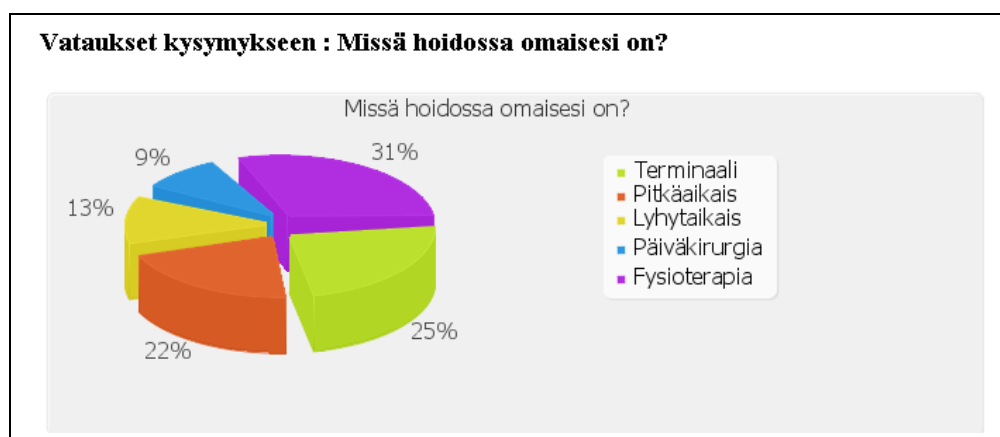


Kuva 7. Monivalintakysymys, jonka vastauksena saadaan serialisoitu vastaus.

Serialisaatio ratkaisi työn yhden keskeisimmistä ongelmista, joka liittyi vastauksiin ja kysymyksien tietojen asettamiseen tietokantaan sekä niiden lukemiseen sieltä.

## 2.6 Kaavioiden piirtokirjastot

Työssä hyödynnettiin kahta erillistä kaavioiden piirtämiseen tarkoitettua kirjastoa, joilla voidaan muodostaa niille annettujen arvojen perusteella kaavioita. Tällaisia kirjastoja Php:lle on saatavilla useita, niin kaupallisia kuin maksuttomiakin, ja niistä työhön valittiin käytettäväksi kirjastot pChart ja libChart niiden maksuttomuuden ja hyvän dokumentaation perusteella. (Trémeaux 2007; PChart 2008.)



Kuva 8. pChart-kirjaston tuottama kaavio.

## 2.7 Captcha

Captcha eli Completely Automated Public Turing test to tell Computers and Humans Apart. On eräs monista keinoista, jolla estetään roskapostittajien mahdollisuus luoda järjestelmään tunnuksia skripteillään ja erilaisilla ohjelmilla. Tekniikka on käytössä esimerkiksi monilla keskustelupalstoilla. Captchan idea on yksinkertainen. Tietokoneohjelmat, jotka luovat automaattisesti tunnuksia tai lähettävät viestejä, eivät kykene tunnistamaan kuvaan tai ääneen piilotettua tekstiä oikein, joka pyydetään syöttämään takaisin ohjelmalle ennen varsinaisen tiedon hyväksymistä. Tämä mahdollistaa yksinkertaisesti tietokoneohjelmilta tulevan datan erottelun käyttäjien syöttämästä datasta. Varsinaisessa työssä käytettiin ainoastaan kuvavarmennetekniikkaa. (Captcha creator 2007.)



Kuva 9. Kuvavarmenne.

## 2.8 Mysql-injektio

Lyhyesti Mysql-injektio on toimenpide, jossa käyttäjä sijoittaa haitallista Sql-koodia suoritettavaan Sql-lausekkeeseen. Yleensä tämä tapahtuu paikassa, jossa käyttäjää pyydetään syöttämään jotain dataa Web-palvelimelle. Se käsitellään Sql-lausekkeen sisällä. Injektointi tapahtuu syöttämällä muuttujan sisään katkaisumerkkejä (\, ', "), jolloin Mysql saadaan tulkitsemaan lauseke käyttäjän haluamaksi lausekkeeksi. Kyseisellä toimenpiteellä vahingollinen käyttäjä voi esimerkiksi pyrkiä vahingoittamaan tietokantaa tai pyrkiä kirjautumaan ohjelmaan pääkäyttäjänä.

Esimerkki Sql-injektioista:

```
<?php
$c = mysql_connect("localhost", "devilish", "xxxxx");
mysql_select_db("devilish", $c);
$_POST['username'] = 'devilish';           //olkoon tunnus devilish
$_POST['password'] = "' OR '='";         //Injektoiva merkkijono
$query = "SELECT * FROM users WHERE user='{$_POST['username']}' AND
password='{$_POST['password']}'";
echo $query;    //Lausekkeen tulostus
?>
```

Tulostus Query-muuttujan sisällöstä:

```
SELECT * FROM users WHERE user='devilish' AND password='' OR ''=''
```

Lausekkeen merkitys on nyt, että valitaan käyttäjän devilish-tiedot millä tahansa salasanalla. Tästä seurauksena aiheutuu vakava tietoturva-aukko, kun OR-operaattori pääsee vuotamaan salasanamuuttujan yli. Ohjelmoijan haluama toiminto olisi ollut valita devilish-käyttäjätiedot salasanan ja tunnuksen täsmätessä. Esimerkissä injektiota voisi hyödyntää ohjelmaan kirjautumisella ilman salasanaa.

Ratkaisuna ongelmaan on käyttää luettaessa tietoja `mysql_real_escape_string`-nimistä funktiota, joka lisää sopivasti katkaisun estäviä merkkejä ennen katkaisumerkkiä. Tällä tavoin voidaan estää injektion syntyminen muodostettavassa Sql-lausekkeessa.

Sql-injektio voidaan estää käyttäen `mysql_real_escape_string` funktiota:

```
<?php
$c = mysql_connect("localhost", "devilish", "xxxxx");
mysql_select_db("devilish", $c);
$_POST['username'] = 'devilish';           //olkoon tunnus devilish
$_POST['password'] = "' OR ''='";         //Injektoiva merkkijono
$query = "SELECT * FROM users WHERE
user='".mysql_real_escape_string($_POST["username"],$c)."' AND pass-
word='".mysql_real_escape_string($_POST["password"], $c)."'";
echo $query;    //Lausekkeen tulostus
?>
```

Tulostus Query-muuttujan sisällöstä:

```
SELECT * FROM users WHERE user='devilish' AND password='\'' OR '\''=\''
```

Tulostuksesta voidaan huomata, ettei OR-operaattori enää pääse ns. vuotamaan ulos salasanamuuttujasta, josta tuloksena on syntynyt turvallinen Sql-kyselylauseke. Näin ollen injektion hyödyntäminen lausekkeessa ei ole enää mahdollista. (Kekkonen 2009; Mysql; Php 2009; Tizag 2008;)



## 2.9 Md5-tiivisteet, Salttaus-menetelmä ja Rijndael

Md5 eli message-digest on yksisuuntainen salausalgoritmi. Yksisuuntaisella algoritmilla tarkoitetaan sitä, ettei luotua salausta voida purkaa, koska kyseessä on alkuperäisen tiedon tiiviste. Md5:n salausidea perustuu algoritmiin, joka tuottaa tiivisteitä hyvin arvaamattomasti. Näin ollen tiivisteestä on hyvin vaikeaa saada selville niiden alkuperäistä sisältöä. Md5 on yksi suosituimmista tavoista salata salasana. Sen hyvänä puolena voidaan pitää sitä, ettei edes järjestelmän ylläpitäjä voi lukea tiivisteiden alkuperäistä sisältöä selväkielisenä. Kuitenkin yleisimmät Md5:n tuottamat tiivisteet ovat nykyään hyvin tunnettuja, ja pelkkä Md5-algoritmin käyttäminen ei riitä näin ollen varmistamaan riittävän hyvää suojaa salasanoille. Internetissä ovat valmiiksi saatavilla useimmin käytettyjen salasanoiden tiivisteet ns. sanakirjoissa, joista on helppoa ja nopeaa etsiä heikkojen tiivisteiden alkuperäistä salasanaa. (Christoph 2004; Rivest 1992.)

Salt-termiä käytetään ohjelmointimaailmassa salasanoiden Md5-tiivisteiden yhteydessä. Se tarkoittaa käytännössä käyttäjän salasanan vahvistamista vahvemerkiksi jollakin toisella merkkijonolla. Menetelmällä pyritään estämään tyypillisimmät sanakirjahyökkäykset, joissa tietokannasta etsitään salasanatiivisteiden vastaparia. Yksinkertaisimmillaan Salt-menetelmää voidaan käyttää, kun käyttäjä syöttää salasan, jonka perään lisätään aina vakio merkkijono lkjdfkjl3fs. Lopputuloksena saadaan alkuperäinen salasana+Salt-avain, joka tuottaa hyvin erilaisen Md5-tiivisteiden kuin pelkkä alkuperäinen annettu salasana. Salt-menetelmä estää tehokkaasti salasan etsimisen ns. sanakirjoista, joihin on koottu valmiiksi usein käyttäjien käyttämiä salasanoiden tiivisteitä. Koko Salt-avaimen toimintaidea perustuu täysin Md5-algoritmin ennalta arvaamattomiin salasanatiivisteisiin. (Christoph 2004.)

Rijndael on nykyään vielä murtamaton salausmenetelmä, joka mahdollistaa 128-, 192- ja 256-bittisen salauksen. Se on yksi tämän päivän tehokkaimmista tiedon salaamiseen käytävistä standardeista. Sen ovat kehittäneet Joan Daemenin ja Vincent Rijmenin. Menetelmästä usein puhutaan synonyymina Aes:n eli Advanced Encryption Standardin kanssa, vaikka kyseessä on Aes-standardia laajentava

menetelmä. Nykyisin. Rijndael täyttää vaatimukset mm. Yhdysvaltain tiedustelupalvelun salaisen tiedon salaamiseksi. (Daemen & Rijmen 1999.)

### 3 OHJELMAN TOTEUTTAMINEN

#### 3.1 Ohjelman suunnittelu

##### 3.1.1 Ohjelman kehitysympäristön valitseminen

Opinnäytetyö toteutettiin käyttäen hyväksi avoimen lähdekoodin ohjelmistoja, koska ne tarjosivat ilmaista alustaa ohjelman kehitykselle ja suurinta yhteensopivuutta muiden järjestelmien kanssa. Tietokantaohjelmistoksi valittiin InnoDB-niminen Mysql:n tietokantalaajennus, koska se kykenee käsittelemään viiteavaimia äiti- ja tytärtaulujen välillä. Toiminnot hallintapaneeli, vastaaminen sekä kyselylomakkeiden arviointi toteutettiin käyttäen Php:tä. Php valittiin työhön, koska tutkimustiedot haluttiin kerätä Web-lomakkeelta. Näin oli myös luonnollista valita muiden osien ohjelmointialustaksi sama ympäristö eheän kokonaisuuden saavuttamiseksi.

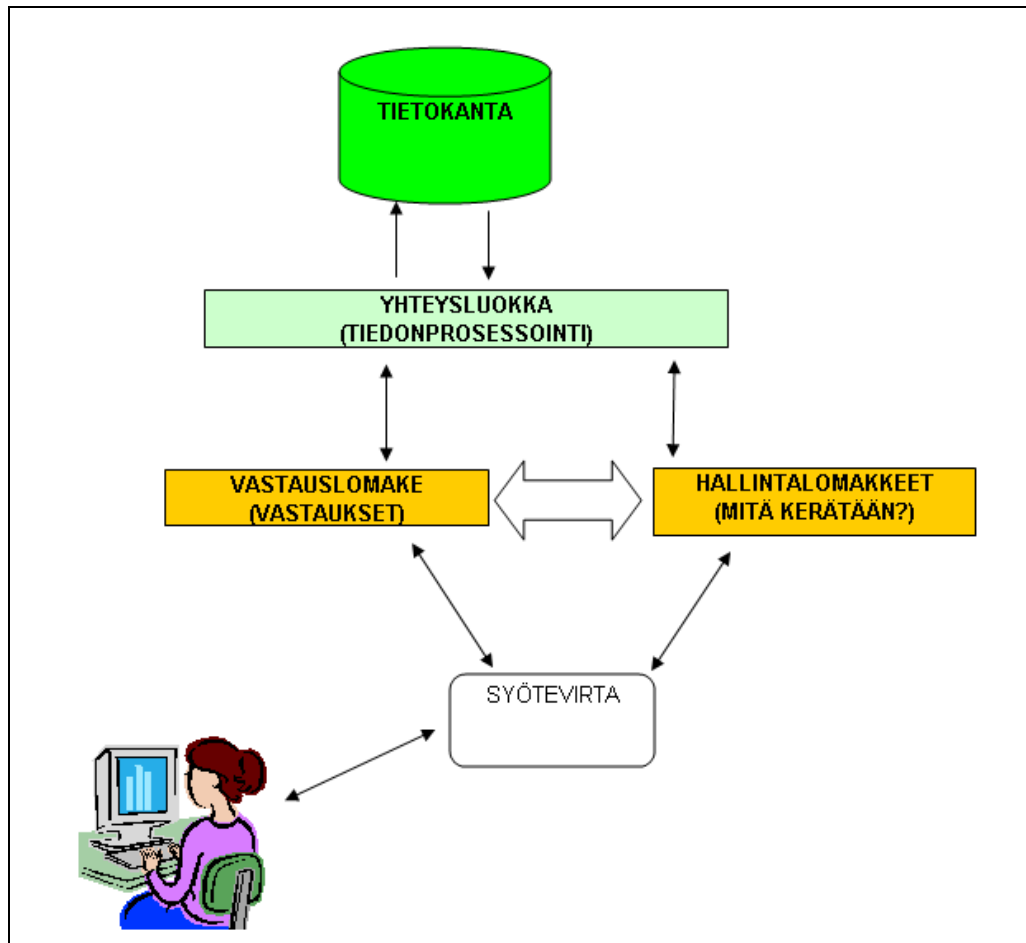
##### 3.1.2 Toteutettavien ominaisuuksien suunnittelu

Lähtökohtana ohjelman ominaisuuksien valinnassa pidettiin, että ohjelmaa voitaisiin käyttää tulevaisuudessa esimerkiksi monikansallisessa yrityksessä työpaikkaviihtyvyyskyselyn suorittamiseen ja sen arviointiin. Ohjelman suunnittelu aloitettiin siltä vaadittavien ominaisuuksien listaamisella käsitekarttaan, jotta tietokannan ja ohjelmarakenteen suunnittelemisen olisi helpompaa.

Tällaisia ominaisuuksia ohjelmassa ovat seuraavat:

- mahdollisuus luoda useita samanaikaisia kyselyitä
- kyselyissä dynaamiset kysymykset
- tuki suorittaa myös anonyymejä kyselyitä
- tuki sähköpostivarmennukselle
- tuki käyttäjäryhmille kyselyissä
- tuki kyselyiden rakentamiseen useilla kielillä
- tuki yleisemmin tunnetuille näppäimistömerkeille
- tuki aikarajoitteisille kyselyille
- tuki useille hallintakäyttäjille
- tuki käyttäjäryhmille hallinnassa
- riittävän turvallinen salasanojen tallennus
- tuki lomakkeiden monikielisyydelle
- tuki ryhmätyöskentelylle kyselyiden parissa
- kenttien automaattinen pituuden skaalaus
- tietojen reaaliaikainen tarkistus.

Ominaisuuksien kartoittamisen jälkeen ohjelma suunniteltiin alhaalta ylöspäin. Seuraavaksi pohdittiin, kuinka ohjelmaan luotavat lomakkeet tulisi jakaa niin, jotta ohjelman rakenteesta saataisiin mahdollisimman yksinkertainen. Ohjelma jaettiin tässä vaiheessa kahteen loogiseen kokonaisuuteen, jotka ovat kyselyiden hallinta ja kyselyihin vastaaminen. Seuraavaksi ohjelmalle hahmoteltiin yhteysluokan eli yhteysolion toimintaperiaate, jonka avulla vastaus- ja hallintalomakkeet pystyvät kommunikoimaan tietokannan kanssa. Oliosunnittelu johti siihen, että ohjelmalle syntyi seuraavanlainen looginen toimintarakenne.



Kuva 10. Ohjelman toiminta.

Toimintaidea lyhyesti:

Käyttäjän vastatessa kyselylomakkeeseen (kuva 10) hän luo syötevirtaa, jonka vastaanottaa vastauslomake. Vastauslomakkeen tehtävänä on käsitellä syötevirta niin, että se voidaan viedä yhteysluokan käsiteltäväksi. Käsittely voi olla esimerkiksi lomakkeelle syötettyjen tietojen oikeaoppisuuden tarkistaminen. Yhteysluokan tehtävänä puolestaan on muodostaa sekä suorittaa Sql-lausekkeet, joilla tietoa voidaan käsitellä ja hakea.

Seuraavaksi suunniteltiin, millaisia osia ohjelmassa tullaan tarvitsemaan sekä miten niitä voitaisiin käyttää mahdollisesti uudelleen jossakin toisessa ohjelman osassa. Suunnittelun helpottamiseksi luotiin käsitekartta ohjelman toiminnoista, joita toteutuksen yhteydessä tarvittiin. Siihen sisältyvät seuraavat osiot:

- kuvavarmennus
- lomakkeiden monikielisyys
- tietokannan ja ohjelman välinen kommunikaatio
- kaaviot
- tietojen salaus ja ohjelmaan kirjautuminen
- sähköpostivarmennus.

### 3.1.3 Sivuston rakenteen suunnitleminen

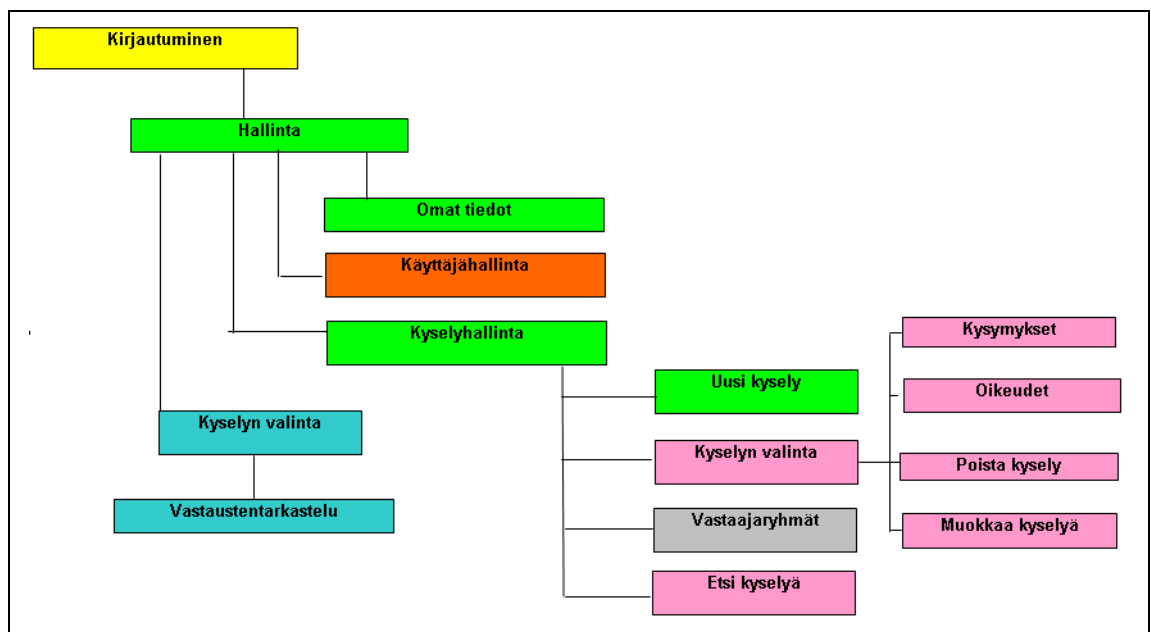
Ohjelman osien loogisen rakenteen suunnittelun jälkeen suunniteltiin ohjelman sivuston rakenne. Luvussa toteutettavien ominaisuuksien suunnittelussa viitattiin kahteen suurempaan kokonaisuuteen sivuston rakenteessa, jotka olivat hallintalomakkeet ja vastauslomakkeet. Nämä ohjelmalohkot toimivat keskeisinä jakajina ohjelmalohkoille, joihin toteutettu ohjelmakoodi sijoitettiin. Vaadittavien ominaisuuksien perusteella hahmoteltiin, että ainakin seuraavia lomakkeita tarvittaisiin ohjelman toteutuksessa:

- vastauslomake
- kyselyiden hallinta
- kysymysten hallinta
- käyttäjien hallinta
- tuloksien tarkastelu
- ryhmien hallinta
- käyttäjätunnuksen perustamislomake.

Lomakkeiden hahmottelun jälkeen pohdittiin, miten nämä lomakkeet pitäisi järjestellä kahteen ohjelmalohkoon loogisen kokonaisuuden saavuttamiseksi. Lomakkeen ohjelmalohkon valinta tehtiin käytännössä miettimällä, oliko lomake ominaisuuksiltaan puhtaasti hallinnallinen vai vastauksia keräävä.

### 3.1.3.1 Hallintasivuston rakenne

Ohjelman hallintasivusto toimii keskeisimpänä osana ohjelmaa, koska sen avulla luodaan niin kyselyt kuin kysymyksetkin ja suoritetaan vastaajilta saatujen vastausten arviointi. Kuvassa 11 on esitetty hallintasivuston lomakerakenne, jonka lomakkeiden käyttöoikeuksiin liittyvät rajoitukset ovat nähtävillä taulukossa 1.



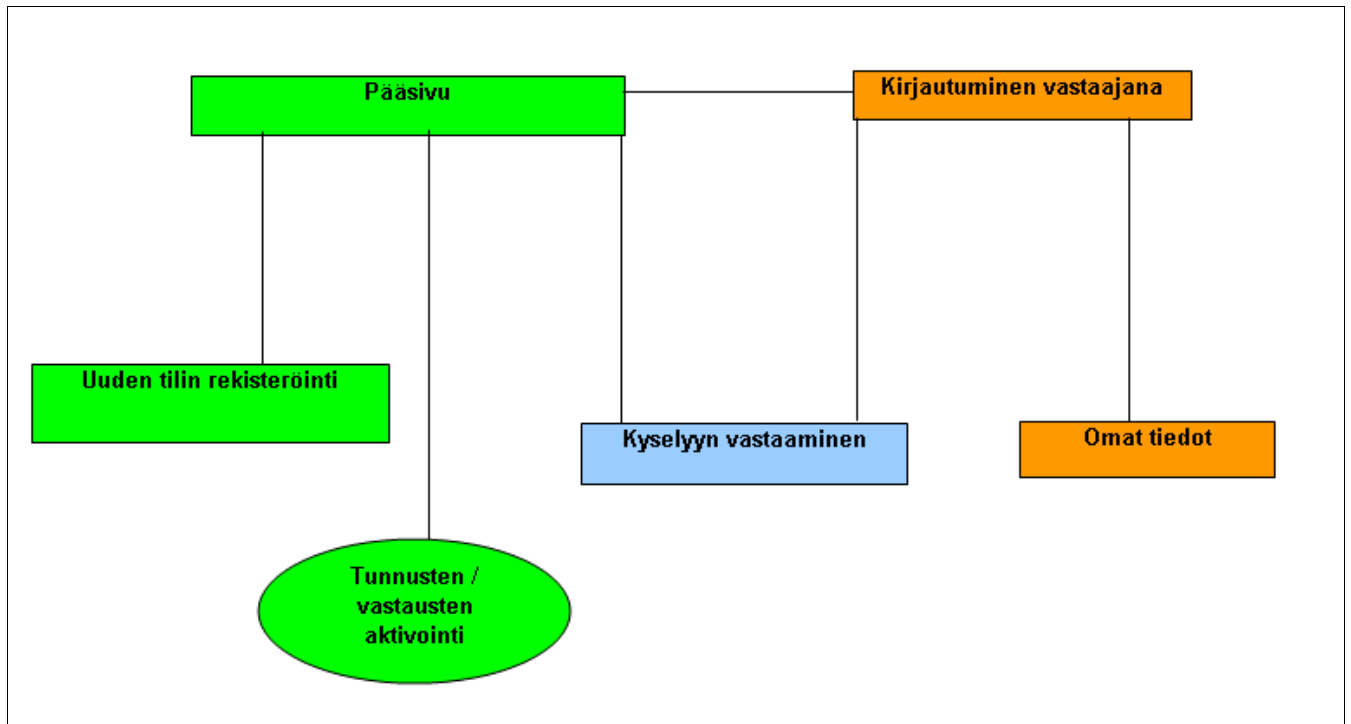
Kuva 11. Hallintasivuston lomakkeet.

Taulukko 1. Hallintasivusto.

Väri	Toiminnot pääkäyttäjälle	Toiminnot käyttäjälle	Rajoitukset
	Kaikille avoin alue	Kaikille avoin alue	-
	Ei rajoituksia	Ei rajoituksia	-
	Ei rajoituksia	Pääsy evätty	-
	Ei rajoituksia	Rajoitettu	Rajoitettu käyttäjiin, joilla on kyselyyn kirjoitus- tai lukuoikeus.
	Ei rajoituksia	Rajoitettu	Tietojensaanti rajoitettu käyttäjiin, joilla on kirjoitusoikeus.
	Ei rajoituksia	Ei rajoituksia ryhmän lisäämiseen. Poistaminen onnistuu jos ryhmä on tyhjä.	-

### 3.1.3.2 Vastaussivuston rakenne

Ohjelmassa toimivan vastaussivuston tehtävänä on huolehtia vastauksien keräämisestä järjestelmään ja huolehtia uusien vastaajien rekisteröinnistä järjestelmään. Kuvassa 12 on näkyvillä vastaussivuston looginen rakenne.



Kuva 12. Vastaussivuston rakenne.

Taulukossa 2 ovat nähtävillä värikoodit, joita on käytetty kuvassa 12 toimintojen selventämiseksi.

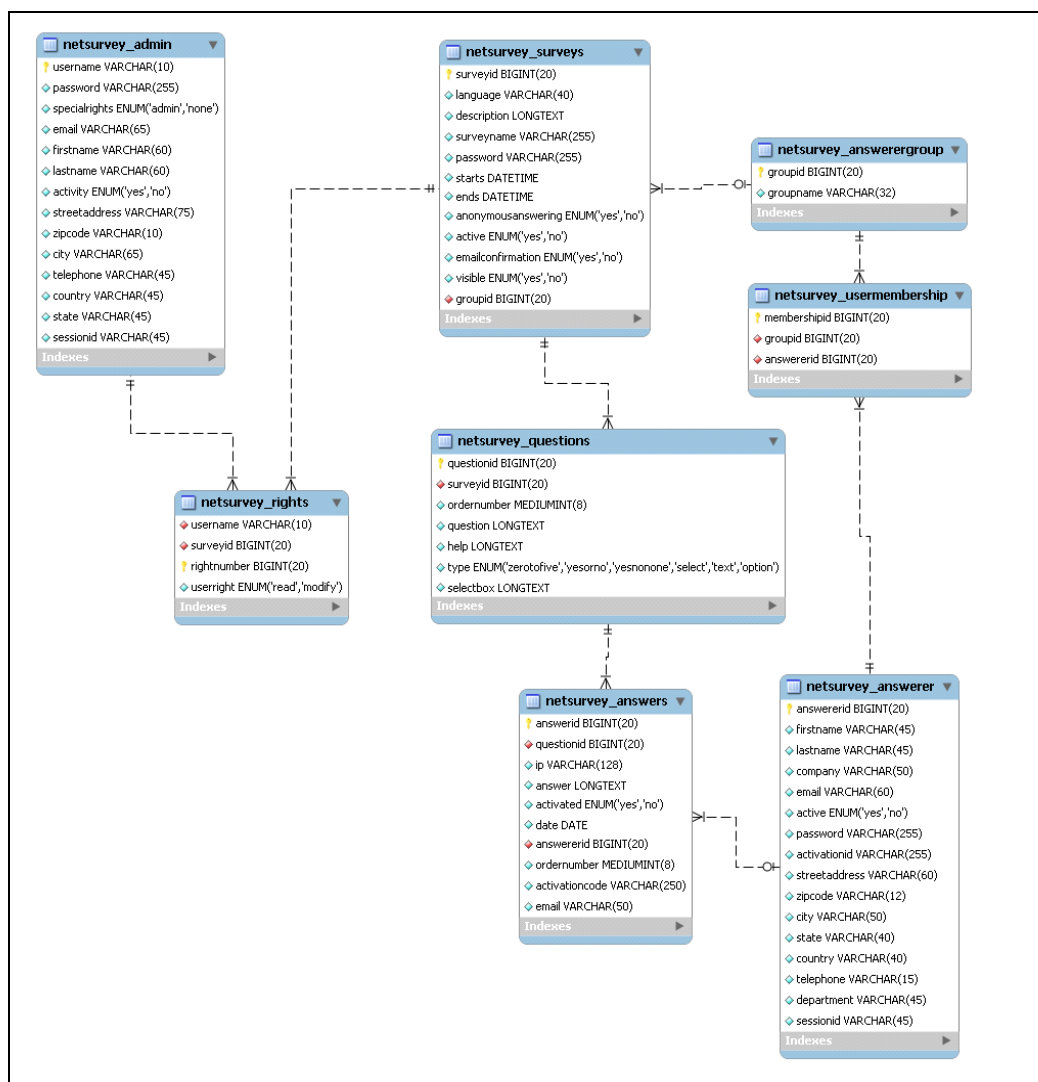


Taulukko 2. Vastaussivusto.

Väri	Toiminnot käyttäjälle	Rajoitukset
	Uuden tilin rekisteröiminen, tunnusten sekä vastausten aktivointi.	Uuden tilin rekisteröiminen voidaan kytkeä pois käytöstä yhteysluokassa. Alueella ei käyttöön liittyviä rajoituksia.
	Kyselyyn vastaaminen	Lomake saatavilla kaikille avoimella alueella, jolloin vastaukset kirjataan anonymieina. Lomake on myös saatavilla kirjautuneille käyttäjille, jolloin vastaukset talletetaan vastaajan tiedoin varustettuna.
	Omien tietojen tarkastelu ja kyselyjen listaus vain kirjautuneille vastaajille.	Alueelle pääsevät vain kirjautuneet käyttäjät.

### 3.1.4 Tietokanta

Ohjelmalle määritettyjen vaatimusten jälkeen tärkein vaihe koko työn toteuttamisen kannalta on tietokannan suunnittelu ja toteutus, koska tietokanta sisältää kaiken tiedon, joka ohjelmalla on käytettävissä. Samalla tietokanta luo perustan koko ohjelmalle sekä kaikille sen sisältämille ominaisuuksille. Voidaan sanoa myös, että hyvin suunniteltu tietokanta vastaa jo puoliksi tehtyä ohjelmaa. Huonosti toteutetulla tietokantasuunnittelulla on samanlainen vaikutus kuin talolla, joka on rakennettu huonojen perustuksien päälle. Vaiheen toteutuksessa käytettiin Mysql Workbench 5.0 -nimistä ohjelmaa, jolla voidaan tarkastella saatua tulosta myös graafisesti näytöltä.



Kuva 13. Tietokannan rakenne.

Tietokannan rakenteeksi valikoitui kuvan 13 rakenne, koska se palveli parhaiten ohjelmaan suunniteltuja ominaisuuksia. Taulukossa 3 on kuvaus jokaisesta taulusta sekä siihen liittyvistä transaktioista. Tuloksena saadut Sql-lausekkeet ovat myös nähtävillä liitteessä.

Taulukko 3. Taulut ja transaktiot kuvauksineen.

Nimi	Sisältö	Vyörytykset
admin	Hallintakäyttäjät	Ei transaktioita
rights	Kyselyiden käyttöoikeudet	Transaktiot tauluihin admin ja surveys. Päivityksessä ja Poista-operaatioissa cascade-toiminta.
surveys	Kyselyt	Transaktio tauluun answerergroup. Päivitettäessä answerergroupia cascade ja Poistossa set null-päivitykset.
questions	Kysymykset	Transaktio tauluun surveys päivitys ja poista operaatioissa cascade-toiminta.
answers	Vastaukset kysymyksiin	Transaktio tauluun questions. Molemmissa operaatioissa cascade-päivitys. Transaktio tauluun answerer. Päivitettäessä cascade ja poistettaessa set null-toiminta avaimelle.
answerer	Rekisteröityneet vastaajat	Ei transaktioita
usermembership	Vastaajille rekisteröidyt vastaajaryhmät	Sisältää transaktion tauluihin answerergroup ja answerer. Päivitys ja Poista-operaatioissa cascade-toiminto.
answerergroup	Vastaajaryhmät	Ei transaktioita

### 3.2 Toteutus

Ohjelman suunnittelun jälkeen seuraavana isona haasteena oli ohjelman toteuttaminen sille määriteltyjen tavoitteiden mukaan. Tätä työvaihetta voidaan pitää työssä identtisenä vaiheena talonrakennusprojektin vaiheelle, jossa rakentamiselle luvat on hankittu sekä piirustukset tehty, mutta varsinaista rakennusurakkaa ei ole vielä aloitettu. Seuraavissa osioissa esitetään ohjelman keskeisimmät lomakkeet niiden tekojärjestyksessä sisältöineen. Lisäksi esitellään lomakkeiden ohjelmoinnissa ilmenneitä tulleita keskeisiä ongelmia ja niissä käytettyjä keskeisiä ratkaisumalleja ja ratkaisuja.

#### 3.2.1 Kyselyhallintaan kirjautuminen

Ohjelman luominen aloitettiin luomalla ensimmäiseksi kirjautumislomake kyselyhallintaan, koska se on ensimmäinen lomake, johon hallintakäyttäjä törmää halutessaan käyttää ohjelmaa. Sen toteuttamisessa oli otettava huomioon useita tietoturvaan liittyviä seikkoja, jotka tarjosivat haasteita toimintojen toteuttamisessa.

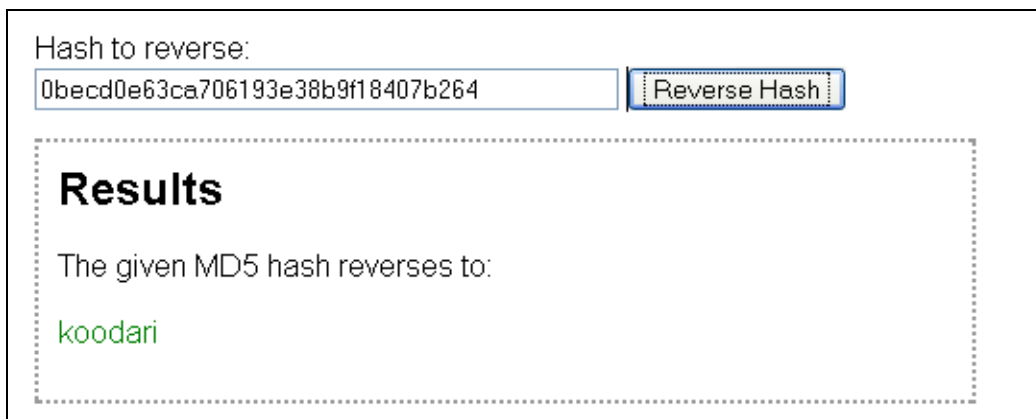


Kuva 14. Kirjautumislomake.

### 3.2.1.1 Keskeiset näkökulmat ja tietoturva

Salasanat:

Kirjautumislomakkeen suunnitteluun liittyy useita tietoturvaongelmia, kuten nykyisin yleisesti käytettävien salausrakenteiden riittämättömyys. Yhtenä riskitekijöistä voidaan pitää salasanojen tallennusta tietokantaan, joka on kuitenkin pakollinen toimenpide. Md5-salasanojen kohdalla riski on helppo todeta esimerkiksi tarkastelemalla sanakirjaa, joista löytyvät yleisimpien salasanojen Md5-tiivisteet:



Hash to reverse:

0becd0e63ca706193e38b9f18407b264

Reverse Hash

**Results**

The given MD5 hash reverses to:

koodari

Kuva 15. MD5-sanakirja. (Reverse Md5 hash lookup.)

Tietokantaan tallennusta varten varsinainen salasana salataan käyttäen kahta erillistä salausrakennetta, Md5:tä ja Rijndaelia. Tämä tapahtuu niin, että varsinaisesta salasanasta ja Salt-avaimesta lasketaan ensin Md5-tiiviste, joka sittemmin salataan käyttämällä 256-bittistä Rijndael-standardia. Tämä estää mahdollisten salasanojen purkamisen tietokannasta jos se kaapattaisiin.

Ohjelman salausfunktion kutsu:

```
$class->saltedpassword(md5(utf8enc($_POST["password"]).$class->getsaltformd5())));
```

Salausfunktio:

```
public function saltedpassword($password) {
    $iv_size = mcrypt_get_iv_size(MCRYPT_RIJNDAEL_256,
    MCRYPT_MODE_ECB);
    $iv = mcrypt_create_iv($iv_size, MCRYPT_RAND);
    $crypttext = mcrypt_encrypt(MCRYPT_RIJNDAEL_256, $this->key, $class->get-
    saltkey(), $password, MCRYPT_MODE_ECB, $iv);
    return base64_encode($crypttext);
}
```

Salasanalle koodari luotu salttaamaton salasanatiiviste:

Md5-tiiviste: 0becd0e63ca706193e38b9f18407b264

Sama tiiviste salattuna Rijndaelilla käyttäen base64-muotoa:

Bwhfmg2NgaX9ZdeRqh55rd3CwNeKSegEmUESQGh7W/Q=

Lisäksi on toinenkin vakava tietoturvaongelma, joka liittyy internetissä salasanan lähettämiseen koneelta palvelimelle. Yleisesti käytössä oleva Http-protokolla ei ole mitenkään salattu ja lähetystilanteessa salasanat kulkevat täysin suojaamatta julkisessa internetissä. Kyseinen ongelma voitiin jo havaita kuvasta 3, jossa lähetävä salasana nähtiin suoraan Http-pakettia tarkastelemalla.

Yksinkertaisesti tämä tarkoittaa sitä, että salasana voidaan paljastaa vain tarkkailemalla liikennettä lähetyksen aikana. Kyseistä menetelmää kutsutaan termillä man-in-the-middle ja se on varsin tehokas tapa selvittää käyttäjän lähettämiä tietoja. Teknisesti tämä menetelmä toimii juuri samalla tavalla kuin puhelinkuuntelu, jossa urkitaan tietoja henkilöiden välisestä keskustelusta. Lisäksi sen toteuttaminen on varsin yksinkertaista käyttäen esimerkiksi verkkoon piilotettua välityspalvelinta myös muita keinoja man-in-the-middle -hyökkäyksen toteuttamiseen on olemassa.



Kuva 16. Man-in-the-middle -hyökkäyksen toimintaperiaate.

Man-in-the-middle -ongelmaan ei juuri voida vaikuttaa sivuston metodien suunnittelulla, koska internetliikenne on yleisesti salaamatonta. Ongelmaan paras ja lähes ainut lääke on käyttää salattua HttPs-protokollaa.

Työssä ongelma ratkaistiin HttPs-protokollalla, jolla varmistettiin, ettei ns. man-in-the-middle -tyyppisen hyökkäyksen suorittaminen onnistu. Lisäksi salasana tallennetaan myös kirjautumisen yhteydessä Sessio-muuttujaan Cookien eli keksin tilasta. Näin varmistetaan, ettei salasanaa siirretä koko ajan yhteyttä pitkin, vaikka käytännössä selain välittääkin Sessio-muuttujaan nimen Cookie-muodossa palvelimelle. Tämän toimintatavan ansiosta salasana on vain palvelimen tiedossa ohjelman käyttämisen aikana.

Yhteysoongelmiin saatu ratkaisu ei kuitenkaan täydellisesti estä esimerkiksi yhteyden kaappaamista väärentämällä Sessio-muuttujaa. Kyseisen hyökkäyksen suorittaminen on erittäin hankalaa HttPs-protokollan salauksen takia. Lisäksi yhteyden kaappaaminen on lähes täysin hyödytöntä salasanan kannalta, koska salasana on tallennettu palvelimelle. (Crypto-Gram 2004.)

Lisäksi erityistä huolta lomakkeella jouduttiin myös kantamaan piilevästä Mysql-injektio vaarasta, jossa haitallinen käyttäjä voisi mahdollisesti yrittää kirjautumista tai tietokannan sotkemista käyttäen hyväksi Sql-lausekkeen heikkoutta.

### 3.2.2 Omat tiedot

Omat tiedot –lomake (kuva 17) sisältää tarkastelu- ja muokkauslomakkeen (kuva 18). Varsinainen ohjelmakoodi on toteutettu poikkeuksellisesti niin hallinta- kuin vastaaja-puolellekin pienin muutoksin. Lomake on myös ainoa lomake, jonka kautta käyttäjä voi muokata omia tietojaan. Haastavinta lomakkeen toteutuksessa olivat sen sisältämät eri Ajax-toiminnot, kuten tietojen automaattinen päivitys lomakkeelta tietokantaan.



**Omat tiedot**

Kenttien avaamiseen tarvitaan salasana!

Käyttäjätunnus	devilish
Salasana	


  

Etinimi	Mika
Sukunimi	Mähönen
Osoite	Presidentinpuistokatu 27 A 10
Postitoimipaikka	Pori
Postinumero	28130
Osavaltio	-
Maa	Finland
Puhelin	+358456704540
Sähköposti	mika.mahonen@viishanke.fi
Salasana	
Salasana uudelleen	

Kuva 17. Omat tiedot -tarkastelulomake.

Tarkastelulomakkeen kirjautumistoiminnon toteutuksessa on hyödyntäen Ajax updater -toimintoa niiden selainten osalta, joissa JavaScript-tuki on saatavilla. Näin ollen käyttäjän ei tarvitse hyväksyä kirjoittamaansa salasanaa painamalla hyväksypainiketta. Selaimissa, joissa ei ole JavaScript-tukea, käyttäjältä edellytetään Hyväksy-painikkeen käyttöä tietueiden lukituksen avaamiseksi.





## Omat tiedot

Etunimi	Mika
Sukunimi	Mähönen
Osoite	Presidentinpuistokatu 27 A 10
Postitoimipaikka	Pori
Postinumero	28130
Osavaltio	-
Maa	Finland ▼
Puhelin	+358456704540
Sähköposti	mika.mahonen@viishanke.fi
Salasana	
Salasana uudelleen	

Kuva 18. Muokkauslomake.

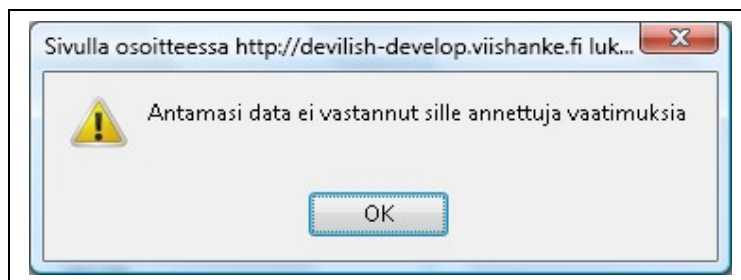
Muokkauslomakkeella käyttäjä pääsee muokkaamaan omia tietojaan lukuun ottamatta käyttäjätunnusta ja käyttöoikeuksia. Selaimissa, joissa on saatavilla JavaScript-tuki, on toteutettu Omat tiedot -lomakkeelle älykkäitä toimintoja, kuten automaattinen tekstin valitseminen, tietojen automaattinen päivitys tietokantaan ja saatujen tietojen tarkastus. (Prototypejs 2009.)

Kaikissa kuvassa 18 esitetyissä lomakkeen kentissä on käytettävissä tekstin automaattinen valinta klikatessa haluttua kenttää, joka voidaan havaita kuvasta 19.



Kuva 19. Automaattinen kentän sisällön valinta.

Käyttäjän muuttaessa kentän sisältöä sen uusi tieto tarkistetaan ja tarkistettu tieto päivitetään tietokantaan. Tilanteessa, jossa käyttäjän antama tieto on virheellistä, palautetaan kentän alkuperäinen tieto takaisin kenttään. Tämän jälkeen käyttäjälle ilmoitetaan annetun syötteen virheellisyydestä JavaScript-alert ilmoituksella.



Kuva 20. Virheellinen syöte.

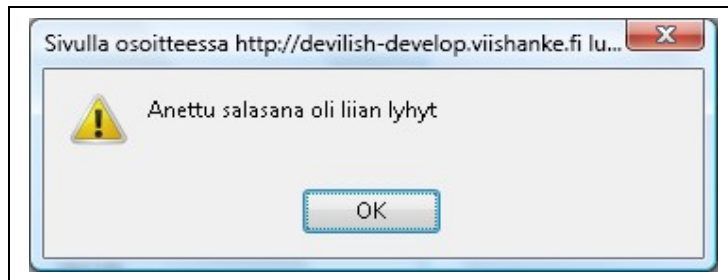
Salasana uudelleen -kentän valintatilanteessa käyttäjä yrittää syöttää salasanan ensiksi Salasana uudelleen -kenttään. Valitaan salasanakenttä käyttäjän puolesta automaattisesti. Toiminnon avulla voidaan välttää päivitystietojen lähettäminen molemmista kentistä. Lisäksi toiminnon avulla voidaan välttää myös kenttien välillä ristiin suoritettavia tarkistuksia.

Salasanakenttien tarkistuksessa käytetty toimintaperiaate on seuraava: tarkistetaan, että salasanakenttä täydennetään ensin. Tämän jälkeen tarkistetaan, että salasana täyttää sille annetun vaatimuksen minimipituudesta, jonka jälkeen voidaan avata salasana uudelleen -kenttä salasanan todennusta varten. Todennuksen onnistuttua näytetään käyttäjälle viesti onnistuneesta salasanan vaihdosta.



Kuva 21. Käyttäjä klikkaa salasana uudelleen -kenttää.

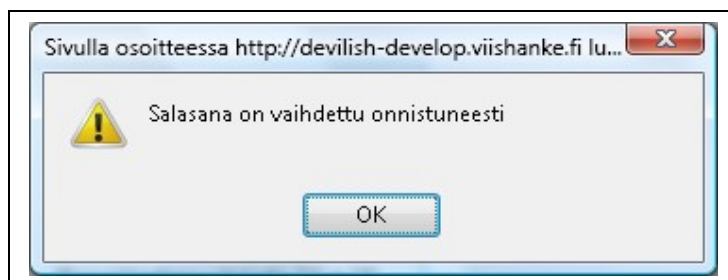
Viestit:



Kuva 22. Liian lyhyt salasana.



Kuva 23. Varmenne ja salasana eivät täsmänneet.




Kuva 24. Salasana vaihdettu onnistuneesti.

Seuraavassa taulukossa ovat näkyvillä kaikki tarkastukset ja toiminnot, jotka liittyvät Omat tiedot -muokkauslomakkeen kenttiin.

Taulukko 4. Omat tiedot -lomakkeen tarkistukset.

<b>Kenttä</b>	<b>Tarkastukset</b>	<b>Päivitys (Kyllä/Ei)</b>
Etunimi	Minimipituus	Kyllä
Sukunimi	Minimipituus	Kyllä
Osoite	Minimipituus	Kyllä
Postinumero	Minimipituus	Kyllä
Postitoimipaikka	Minimipituus	Kyllä
Osavaltio	Minimipituus	Kyllä
Maa	Minimipituus	Kyllä
Puhelin	Minimipituus	Kyllä
Sähköposti	Sähköpostiosoitteen tarkastus (domain-nimi ja päätte @-merkin takana, vain yksi @-merkki ja merkkijono ennen @-merkkiä)	Kyllä
Salasana	Minimipituus	Ei
Salasana uudelleen	Sama kuin salasananakentän sisältö	Kyllä

Alla olevassa esimerkissä on nähtävillä etunimikentän päivittämiseen ja tarkastamiseen käytetty ohjelmakoodi:


 A screenshot of a web browser showing a text input field. The field has a light blue border and a light blue background. Inside the field, the text 'Mika' is visible. To the left of the field, the label 'Etunimi' is displayed in a small, dark font.

Kuva 25. Näkyvillä input-elementti.

Input-elementin koodi:

```
<input class="readandwrite" type="text" name="firstname" value="<?php echo
(mysql_result($mysql, 0, "firstname")); ?>" id="firstname" onfocus="writein( 'first-
name' )" onblur="writeout( 'firstname' );" onchange="if((document.getElementById('first-
name').value).length >= <?php echo $minl["firstname"]; ?>) { modi-
fypersonaldata('firstname',document.getElementById('firstname').value) } else
{ firstname.value='<?php echo addslashes(mysql_result($mysql, 0,
"firstname")); ?>'; alert ('<?php echo msg("doesntmeetrequirements"); ?>'); } "
maxlength="<?php echo $lenghts["firstname"]; ?>" size="25" onclick="document.-
getElementById('firstname').select();" />
```

Koodissa kentän vähimmäispituus määräytyy minl-taulukkomuuttujan mukaan, jonka sisältö tulostetaan JavaScript-koodin sekaan lomakkeen luomisen yhteydessä. Näin minimipituus on täysin säädeltävissä othersettings-nimisestä luokasta käsin. Kenttien vähimmäispituudet ovat määriteltynä myös othersettings-luokassa, jonka ansiosta kentän minimipituus on säädettävissä dynaamisesti riippuen käyttäjän tarpeista.

Edellisen koodin lisäksi tarvitaan kutsuttava Ajax-päivitysfunktio, jonka nimi on modifypersonaldata:







```
function modifypersonaldata(field, data) {
    new Ajax.Request('page.php?pageid=100', {method: 'post', parameters:{ field:
field, data: data }});
}
```

Esimerkissä olevaa funktiota käytetään tietojen päivitykseen, joka hoitaa tiedon lähetyksen oikeaan osoitteeseen palvelimelle.

### 3.2.3 Käyttäjähallinta

Käyttäjähallinta on osa, jonka avulla ylläpidetään kyselyn hallintajärjestelmään kirjautuvien käyttäjien tietoja. Tämän lomakkeen toimintoja tarvitaan aina, kun ohjelman pääkäyttäjä haluaa tehdä muutoksia järjestelmässä oleviin muihin käyttäjätileihin. Lomakkeella keskeisimmät toteutusongelmat liittyivät salasanojen muuttamiseen, omien tietojen muokkaamiseen ja luvattomien käyttäjien pääsyn estämiseen lomakeelle ja sen toimintoihin.

Käyttäjähallintanäkymästä löytyy käyttäjien esikatselu, poista, muokkaa ja lisää käyttäjätili.

Käyttäjähallinta						
Etunimi ▼▲	Sukunimi ▼▲	Sähköposti ▼▲	Käyttäjätunnus ▼▲	Käyttäjätaso ▼▲	Tilin tila ▼▲	Toimenpide
Mika	Mähönen	mika.mahonen@kasikirja.net	mikamah	Käyttäjä	Toiminnassa	  
Taru	Mähönen	taru.mahonen@vaalijala.fi	mahonta	Käyttäjä	Toiminnassa	  
						2/2

Kuva 26. Käyttäjähallintanäkymä.

### Luo uusi käyttäjätili

Käyttäjätunnus	<input type="text"/>
Etunimi	<input type="text"/>
Sukunimi	<input type="text"/>
Osoite	<input type="text"/>
Postinumero	<input type="text"/>
Postitoimipaikka	<input type="text"/>
Osavaltio	<input type="text"/>
Maa	<input type="text" value="▼"/>
Puhelin	<input type="text"/>
Sähköposti	<input type="text"/>
Salasana	<input type="password"/>
Käyttäjätaso	Pääkäyttäjä <input type="radio"/> Käyttäjä <input checked="" type="radio"/>
Tilin tila	Toiminnassa <input checked="" type="radio"/> Poistettu käytöstä <input type="radio"/>
<input type="button" value="Tyhjennä"/> <input type="button" value="Luo käyttäjätili"/>	

Kuva 27. Uuden käyttäjätilin luominen.

### 3.2.4 Lisää kysely

Toimintoa käytetään uuden kyselylomakkeen luomiseksi tietokantaan. Toiminto vastaa reaali maailmassa toimenpidettä, jossa kyselyn suunnittelija hakee tyhjän paperin kirjoittaakseen siihen ylös kyselyn nimen sekä sen perustiedot. Merkittävimmät ongelmat, joita lomakkeen toteuttamisessa jouduttiin ratkaisemaan liittyvät tarkistuksiin sekä kalenterin käyttöönottoon ohjelmassa.

### Uuden kyselyn luomisvelho

Nimi


Kieli

Suomen kieli (Finnish) ▼


Kuvaus

Salasana

Avautuu

2009-03-11 10:00 

Sulkeutuu



Anonyymi vastaaminen

Käytössä ☐ Pois käytöstä ☒

Sähköpostivarmennus

Toiminnassa ☒ Poistettu käytöstä ☐

Nähtävillä

Kyllä ☒ Ei ☐

Osallistujaryhmä

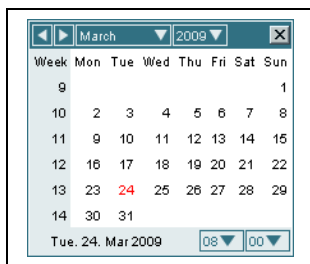
▼

Tyhjennä

Luo kysely

Kuva 28. Lisää kyselylomake.

Kalenteriksi ohjelmaan valittiin Dhtmlgoodies-sivustolla vapaasti saatavilla oleva kalenteri. Se sisälsi kaikki keskeisimmät ominaisuudet, joita ajan valitsemisessa tarvittiin. Keskeinen ongelma kalenterissa ovat selaimet, jotka eivät tue JavaScriptiä. Kuitenkin tällaisia selaimia on käytännössä hyvin vähän, eikä asiasta muodostu ongelmaa ohjelmaa käyttäville käyttäjille. (Dhtmlgoodies 2008.)



Kuva 29. Dhtmlgoodiesin Javascript-kalenteri. (Dhtmlgoodies 2008.)

### 3.2.5 Etsi kyselyä

Etsi kysely -toimintoa voidaan käyttää saatavilla olevien kyselyiden tarkasteluun sekä kyselyiden etsimiseen. Ohjelman peruskäyttäjille se on myös yksinkertainen tapa selvittää, millaiset käyttöoikeudet kyselyyn ovat käytettävissä. Toiminto sisältää kyselysuodattimen sekä kyselyiden näyttöosan, johon suodatuksen jälkeen saadut kyselyt tulostetaan. Näyttöosasta löytyvät mahdollisuudet kyselyiden järjestämiseen laskevaan tai nousevaan järjestykseen. Etsi kyselyä -lomaketta on käytetty myös kyselyn valintalomakkeen pohjana, jossa käyttäjää pyydetään valitsemaan kysely johon suoritettava toimenpide kohdistetaan. Lomakkeelle asetetut järjestely- ja suodatusominaisuudet tarjosivat haastetta ja pohdittavaa toteuttamisen kannalta.

Oikeudet	Kyselyn nimi ▼▲	Vastaajaryhmä ▼▲	Alkaa ▼▲	Loppuu ▼▲	Kieli ▼▲	Käynnissä ▼▲	Kuvaus
Paakäyttö	KOULUTERVEYS		2009-03-16 08:00:00		Finnish	Kyllä	HEI! Teemme tutkimusta kouluvihtyvyydestä, koulujen terveyskasvatuksesta ja op
Paakäyttö	Muutetaan kyselyn kuvausta		2009-03-16 12:00:00	2011-03-16 08:00:00	Finnish	Kyllä	Tähän tulee, joku hieno kuvaus millaisesta kyselystä on oikein todellisuudessa
Paakäyttö	Joku kysely		2009-03-16 13:00:00		Finnish	Kyllä	Tähän tulee hieno kuvaus kyselystä
							3/3

Kuva 29. Pääkäyttäjän näkymä.

Oikeudet	Kyselyn nimi ▼▲	Vastaajaryhmä ▼▲	Alkaa ▼▲	Loppuu ▼▲	Kieli ▼▲	Käynnissä ▼▲	Kuvaus
Vain luku	KOULUTERVEYS		2009-03-16 08:00:00		Finnish	Kyllä	HEI! Teemme tutkimusta kouluvihtyvyydestä, koulujen terveyskasvatuksesta ja op
Muokkaa	Kehitettävää				Finnish	Ei	
							2/2

Kuva 30. Käyttäjän näkymä.



**Suodatukset**

Kyselyn nimi

Kieli

Avautuu

Sulkeutuu

Kyselyn aktiivisuus Kyllä ☐ Ei ☐

Anonyymi vastaaminen Käytössä ☐ Pois käytöstä ☐

Sähköpostivarmennus Toiminnassa ☐ Poistettu ☐

Nähtävillä Kyllä ☐ Ei ☐

Maaliskuu 2009

Viiikko	Ma	Ti	Ke	To	Pe	La	Su
9							1
10	2	3	4	5	6	7	8
11	9	10	11	12	13	14	15
12	16	17	18	19	20	21	22
13	23	24	25	26	27	28	29
14	30	31					

Tänään La, 7. Mai 2009 08:00

Kuva 31. Kyselysuodatin.

### 3.2.6 Muokkaa kyselyä

Lomake toimii ohjelmassa kyselyn viimeistely- ja käyttöönottolomakkeena. Lomakkeen suunnittelussa jouduttiin miettimään tarkasti ongelmia, kuten voidaanko kyselyä muokata sen käyttöönottamisen jälkeen ja jos voidaan niin millä rajoituksilla. Tämä ongelma muodostui lomakkeen toimintojen yhdeksi keskeisimmistä kysymyksistä ohjelmointiratkaisun kannalta, koska ohjelmassa haluttiin välttää tilanteet, joissa kerätty aineisto voisi vääristyä. Vääristymä voisi tapahtua esimerkiksi, jos kysymyksiä voitaisiin muokata kyselyn aloittamisen jälkeen. Ongelma voidaan ratkaista tutkimalla, onko kyselyyn saatua tutkimusaineistoa olemassa.

**Muokkaa kyselyä**

Nimi

Kieli

Kuvaus

Salasana

Avautuu

Sulkeutuu

Anonyymi vastaaminen Käytössä ☐ Pois käytöstä ☐

Sähköpostivarmennus Toiminnassa ☐ Poistettu käytöstä ☒

Nähtävillä Kyllä ☒ Ei ☐

Kuva 32. Muokkaa kyselyä.

### 3.2.7 Muokkaa kyselyn käyttöoikeuksia

Käyttöoikeuksien määrittäminen kyselyyn kuuluu ohjelmassa toteutettuun mahdollisuuteen, jossa useampi käyttäjä voi käsitellä kyselyitä. Lomakkeen käyttäminen on tarpeen tilanteissa, joissa useamman käyttäjän tarvitsee tarkastella tai muokata kyselyyn kuuluvia tietoja. Tällöin tarvitaan myös kyselykohtaisia käyttöoikeuksia, jotta kyselyä tarkastelevien tai muokkaavien käyttäjien määrää voidaan rajata tarpeiden mukaan. Eivätkä kyselyiden tiedot ole kaikkien hallintaa käyttävien käyttäjien saatavilla. Toteutuksen keskeisin haaste oli pystyä rajamaan käyttäjät tarkasti kyselyihin liittyvien oikeuksien perusteella. Lomake itsessään muodostuu kahdesta osasta, muokattavan kyselyn valinnasta sekä käyttöoikeuksien käsittelylomakkeesta.

Valitse	Kyselyn nimi ▼▲	Vastaajaryhmä ▼▲	Alkaa ▼▲	Loppuu ▼▲	Kieli ▼▲	Käynnissä ▼▲	Kuvaus
<input type="radio"/>	Kehitettävää				Finnish	Ei	
							1/1

Kuva 33. Muokattavan kyselyn valinta.

**Kyselyn oikeudet**

Käyttäjätunnus	Oikeudet	Toimenpide
devilish	Vain luku	✗

**Lisää käyttäjälle oikeus kyselyyn**

☐ Muokkaa
 ☒ Vain luku

Kuva 34. Käyttöoikeuksien käsittelylomake.

### 3.2.8 Poista kysely

Poista kysely -lomakkeen avulla käyttäjät voivat siivota vanhentuneet tiedot pois tietokannasta. Se auttaa käyttäjiä ylläpitämään tietokannassa vain tarpeellisen määrän tietoa. Keskeisin ongelma lomakkeella ratkaistiin jo tietokannan suunnitteluvaiheessa, jossa valittiin käyttöön transaktionaalinen tietokanta. Tämä poisti lomakkeen suurimman ja lähes ainoan ongelman eli tiedon eheyden säilyttämisen tytärtauluissa. Suoritetut muutokset lomakkeella vaikuttavat juuri transaktionaalisuuden kautta tauluihin rights, answers ja questions. Esimerkiksi kyselyn poistamisen yhteydessä transaktionaalinen tietokanta huolehtii itse tiedon eheydestä tytärtauluissa. Relationalisessa mallissa tietokannassa jouduttaisiin suorittamaan joukko erilaisia Sql-lausekkeita varmistamiseksi, että tytärtaulujen tiedot ovat tuhottu asianmukaisesti eivätkä ne jää roikkumaan tietokantaan ja häiritsemään mahdollisia uusia luotavia kyselyitä.

Poistettavissa olevat kyselyt						
Valitse	Kyselyn nimi ▼▲	Alkaa ▼▲	Loppuu ▼▲	Kieli ▼▲	Käynnissä ▼▲	Kuvaus
<input type="checkbox"/>	KOULUTERVEYS	2009-03-16 08:00:00		Finnish	Kyllä	HEI! Teemme tutkimusta kouluviihtyvyydestä, koulujen terveyskasvatuksesta ja op
<input type="checkbox"/>	Muutetaan kyselyn kuvausta			Finnish	Kyllä	Tähän tulee, joku hieno kuvaus millaisesta kyselystä on oikein todellisuudessa
<input type="checkbox"/>	Joku kysely	2009-03-16 13:00:00		Finnish	Kyllä	Tähän tulee hieno kuvaus kyselystä
<input type="checkbox"/>	Kehitettävää			Finnish	Kyllä	
<input type="checkbox"/>	Ohjaustarvetesti ammattikorkeakouluihin	2009-03-21 13:05:00		Finnish	Kyllä	Ohjaustarvemittarin avulla pyritään selvittämään, mitä sinä ajattelet koul
						5/7 ➡

Kuva 35. Kyselyn poistolomake.

### 3.2.9 Muokkaa kyselyn kysymyksiä

Muokkaa kyselyn kysymyksiä -lomaketta käytetään kysymyksien luomiseen ja muokkaamiseen. Tarkastelulomakkeelta löytyvät toiminnot kysymyksien lisäämiseen, muokkaamiseen, tarkasteluun ja järjestyksen muuttamiseen. Lomakkeen toteuttamisessa haastavia toimintoja on mm. kysymyksen järjestysnumeron muokkaaminen.

Valitse	Kysymys	Vastausohje	Kysymyksen muoto	Järjestys	Toimenpide
<input type="checkbox"/>	Kerro itsestäsi	jooh	Valintapainikkeet		
<input type="checkbox"/>	Minkälaisen arvosanan antaisit tälle ohjelmalle?		Kyllä / Ei / En tiedä		
<input type="checkbox"/>	Toimiiiko tämä nyt paremmin kuin edellinen		Teksti		
<input type="checkbox"/>	Testi	Dataa	Valintapainikkeet		
					<a href="#">Lisää uusi kysymys</a>

Kuva 36. Muokkaa kyselyn kysymyksiä -lomake.

Muokkaustoiminolla lomakkeella on mahdollista täydentää jo luodun kysymyksen tietoja. Näin käyttäjä voi lisätä esimerkiksi uusia vastausvaihtoehtoja tai muokata olemassa olevan kysymyksen muotoa.

Kysymys

Mielipiteeni ammasteista vaihtuvat vähän väliä.

Vastausohje

Kysymyksen muoto

☒ Valinta vaihtoehtoista  
☐ Valintaruudut

Vaihtoehto 1

Täysin eri mieltä

Vaihtoehto 2

Eri mieltä

Vaihtoehto 3

Samaa mieltä

Vaihtoehto 4

Täysin samaa mieltä

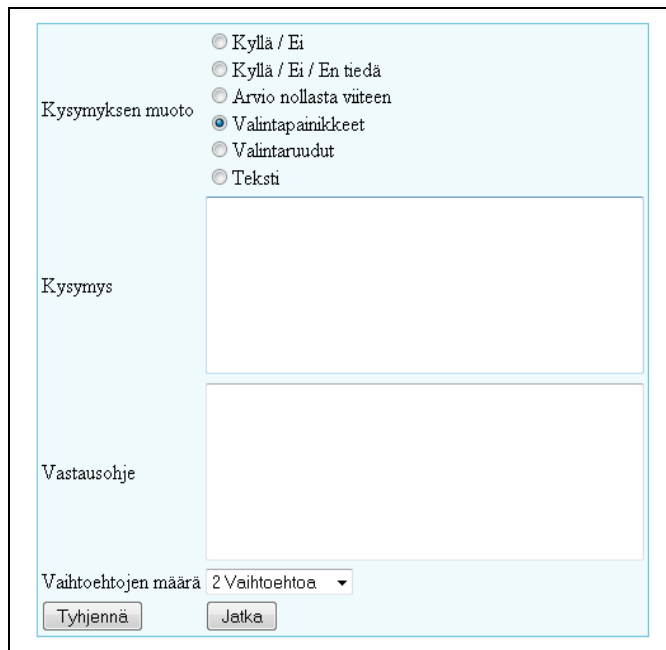
Lisää uusia vaihtoehtoja

Palauta

Talenna muutokset

Kuva 37. Muokkaa kysymystä –toiminto.

Lisää uusi kysymystoiminto löytyy tarkastelulomakkeen oikeasta alakulmasta (kuva 36.) Se sisältää vaihtoehdot kuuden erilaisen kysymystyyppin luomiseksi ohjelmaan.







Kuva 38. Lisää uusi kysymys.

Kysymyksen järjestyksen muuttamiseen käytetyssä toimintamallissa haetaan muistiin nykyinen järjestysnumero, jota sanotaan esimerkissä x:ksi. Tämän jälkeen haetaan muistiin myös naapurikysymyksen numero, jota kutsutaan y:ksi. Seuraavaksi vaihdetaan muutettavan kysymyksen järjestysnumero nolllaksi, jonka jälkeen päivitetään y:n kysymyksen numero x:ksi ja x:n arvo y:ksi.

### 3.2.10 Vastaajaryhmät

Ominaisuus kuuluu ohjelmassa toteutettuun kokonaisuutteen, jossa kysely voidaan suorittaa tietyille joukolla henkilöitä. Tällainen tarve voi esiintyä esimerkiksi, jos kysely halutaan suorittaa vain tietyille yrityksen osastolle. Ominaisuuden avulla on mahdollista valita kaikki ne vastaajat, jotka voivat vastata kyselyyn. Esitellyn ominaisuuden käyttäminen edellyttää olemassa olevaa vastaajatunnusta järjestelmässä.


**Vastaajaryhmien hallinta**

Vastaajaryhmä	Toimenpide
IT05R2	 
IT07R2	 

Lisää uusi vastaajaryhmä

Kuva 39. Vastaajaryhmien hallinta.

Kuvassa 39 on näkyvillä ryhmästä poisto- ja liittämiseen tarkoitettu toiminto. Sen avulla voidaan ryhmään liittää vaivattomasti uusia käyttäjiä tai poistaa vanhoja.

<b>IT07R2</b>			
<b>Lisää tai poista vastaajaryhmän jäseniä</b>			
Käyttäjätunnus	Etunimi	Sukunimi	Toimenpide
mika.mahonen@viishanke.fi	Mika	Mahönen	

Kuva 40. Vastaajaryhmän jäsenet.

### 3.2.11 Kyselyyn vastaaminen

Voidaan sanoa, että kyselylomake on ohjelman eräs keskeisimmistä osista, koska sen avulla kerätään kaikki kyselyyn liittyvä aineisto. Lomakkeen kehittämisessä jouduttiin ratkaisemaan monia ohjelman keskeisiä ongelmia, kuten Http:n tilattomuusongelma sivua vaihdettaessa, jonka seurauksena annetut vastaukset häviävät. Muita merkittäviä haasteita toteutuksen kannalta olivat serialisoituvan vastauksen käsittely sekä erinäiset suojaukset, kuten salasana ja sähköpostivarmennus.

NetSurvey - Mozilla Firefox

http://devilsh-develop.viishanke.fi/survey/index.php?pageid=answeranonsurvey&id=43

Viishanke.fi / loca... NetSurvey Inbox: 25 messa... NetSurvey NetSurvey Plenware Rauma... käyttäjätit - Goo... salasana käyttä...

osoittamat menetelmät. Ammattimaisesti koottu tutkimustiedon ja validien analyysien pohjalta voidaan tehdä riskittömämpiä päätöksiä.

**Kysymykset**

Numero	Apua vastaamiseen	Kysymys	Vastauslomake
1 / 8		Pidätkö nettisivuistamme	Kyllä Ei <input type="radio"/> <input type="radio"/>
2 / 8	?	Miten hyvin tuotteemme vastaa odotuksiasi?	0 1 2 3 4 5 <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>
3 / 8		Aiotko ostaa tuotteitamme uudelleen	Kyllä Ei <input type="radio"/> <input type="radio"/>
4 / 8		Mitä tuotteistamme olet ostanut	<input type="checkbox"/> Domain-hosting <input type="checkbox"/> Lumikola <input type="checkbox"/> Lapio <input type="checkbox"/> Säteilysuoja
5 / 8		Onko tuotteemme sinusta markkinoiden paras?	Ei vastausta Kyllä Ei <input type="radio"/> <input type="radio"/> <input type="radio"/>

Seuraava sivu >>

Tyhjennä

Done

Kuva 41. Kyselyyn vastaamislomake.

Tilattomuusongelma lomakkeella ratkaistiin tallentamalla annetut vastaukset sessio-muuttujaan, joka on palvelimen muistissa. Sessio-muuttujaan vastausten tallentaminen tapahtuu hyödyntäen Ajaxista tuttua automaattista lähetä-toimintoa. Näin kyselyyn vastaajan ei tarvitse huolehtia annettujen vastauksen tallentamisesta ennen sivun vaihtoa. Niiden vastaajien osalta, joiden selaimet eivät tue JavaScriptiä näytetään erillinen Tallennus-painike jokaisen sivun lopussa.



### 3.2.12 Tuloksien tarkastelu

Tuloksien tarkastelu on ohjelman yksi keskeisimmistä lomakkeista, koska sen tehtävä on käsitellä kerätty aineisto sellaiseen muotoon, jota jokainen meistä kykenee ymmärtämään. Tämä käytännössä tarkoittaa graafisia kaavioita sekä diagrammeja, joista tieto on yksinkertaisesti luettavissa. Tarkastelulomake sisältää kyselyn ja kysymyksen valintalomakkeen. Kyselyn valintalomakkeen toiminnan erikoisuutena on se, ettei kyselyitä, joihin ei ole vastattu, myöskään voida näyttää käyttäjille. Haasteellista tarkastelun toteuttamisessa oli libChart- ja pChart-piirtofunktioiden käytön opetteleminen. (PChart 2008; Trémeaux. 2007.)

Valitse	Kyselyn nimi ▼▲	Vastaajaryhmä ▼▲	Alkaa ▼▲	Loppuu ▼▲	Kieli ▼▲	Kuvaus
<input type="radio"/>	KOULUTERVEYS		2009-03-16 08:00:00		Finnish	HEI! Teemme tutkimusta kouluviihtyvyydestä, koulujen terveyskasvatuksesta ja op
<input type="radio"/>	Joku kysely		2009-03-16 13:00:00		Finnish	Tähän tulee hieno kuvaus kyselystä
<input type="radio"/>	Kehitettävää	IT07R2			Finnish	
						3/3

Kuva 42. Tarkasteltavan kyselyn valitseminen.

Käyttäjän valittua kyselyn, jota hän haluaa tarkastella, siirrytään lomakkeelle, jossa ovat listattuina kaikki kyselyyn liittyvät kysymykset ja tarkastelumahdollisuudet. Kysymyksen valintalomakkeen tehtävänä on toimia esivalitsimena sille tiedolle, mitä tietokannasta tuodaan esille.

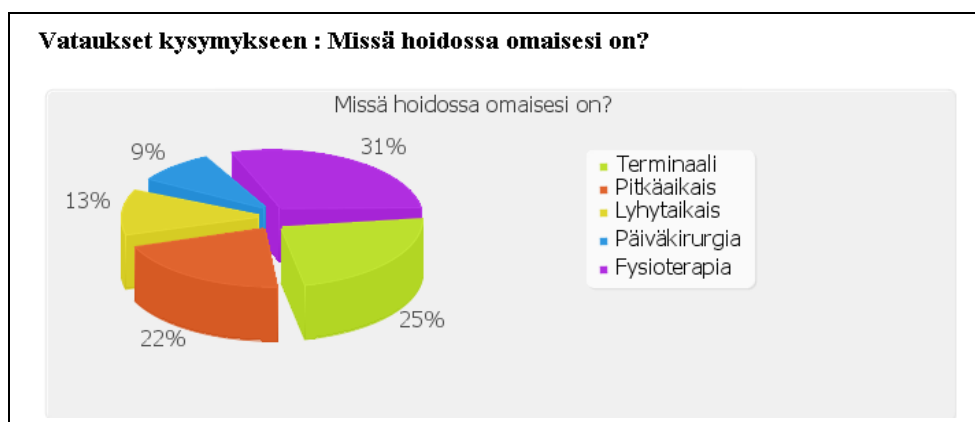
**Kyselyn kysymykset**

Valitse kysymykset joista haluat analyysin tehtävän

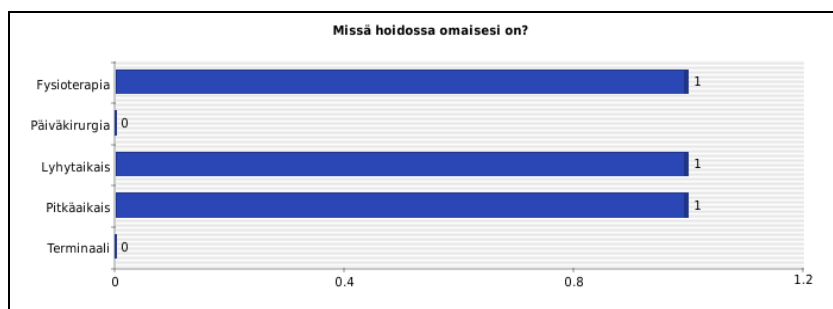
Valitse	Kaavio	Kysymys	Vastausohje	Kysymyksen muoto
<input type="checkbox"/>	<input type="radio"/> Ympyrädiagrammi <input type="radio"/> Pylväsdiagrammi <input checked="" type="radio"/> Molemmat	Sukupuoli		Valinta vaihtoehtoista
<input type="checkbox"/>		Mikä on pituutesi		Teksti
<input type="checkbox"/>		Paino		Teksti
<input type="checkbox"/>	<input type="radio"/> Ympyrädiagrammi <input type="radio"/> Pylväsdiagrammi <input checked="" type="radio"/> Molemmat	Opettajat rohkaisevat minua ilmaisemaan oman mielipiteeni oppitunneilla		Valinta vaihtoehtoista
<input type="checkbox"/>	<input type="radio"/> Ympyrädiagrammi <input type="radio"/> Pylväsdiagrammi <input checked="" type="radio"/> Molemmat	Opettajat ovat kiinnostuneita siitä, mitä minulle kuuluu		Valinta vaihtoehtoista
<input type="checkbox"/>	<input type="radio"/> Ympyrädiagrammi <input type="radio"/> Pylväsdiagrammi <input checked="" type="radio"/> Molemmat	Opettajani odottavat minulta liikaa koulussa		Valinta vaihtoehtoista

Kuva 43. Tarkasteltavan kysymyksen valitseminen.

Ohjelman tavoitteen voidaan sanoa olleen saavutettu, kun tutkimuslomake kykeni antamaan käyttäjälle tietoja, jotka on kerätty kyselyihin vastanneilta käyttäjiltä.



Kuva 44. Pchartilla piirretty piirakkakaavio. (PChart 2008.)

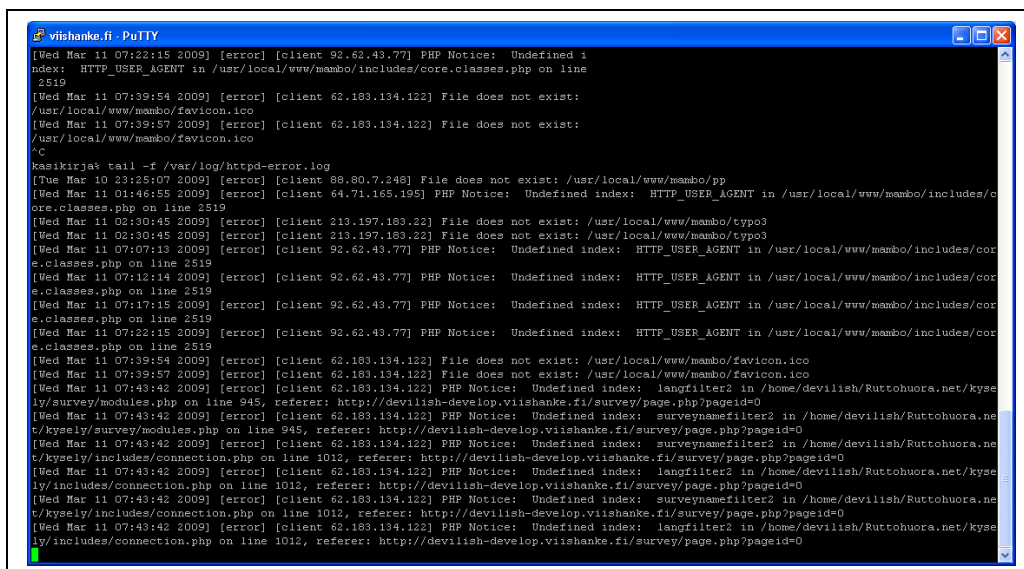


Kuva 45. LibChartilla piirretty pylväsdiagrammi. (Trémeaux 2007.)

### 3.3 Virheiden etsintä ja korjaus

Ohjelman kannalta viimeisin vaihe, muttei vähäisin, oli ohjelmistotestaus. Sen aikana ohjelman toiminnot tarkastettiin etsien virheitä ohjelmoidusta koodista. Vaiheen aikana etsittyjä virheitä olivat alustamattomat muuttujat, varsinaiset suunnittelelmattomat toiminnot ohjelmakoodissa sekä Xhtml-sivujen muotoiluvirheet.

Syntaksivirheiden ja alustamattomien muuttujien etsintä Php-koodista suoritettiin tarkastelemalla Web-palvelimelta saatavia viestejä.



```

viishanke.fi PuTTY
[Wed Mar 11 07:22:15 2009] [error] [client 92.62.43.77] PHP Notice: Undefined index: HTTP_USER_AGENT in /usr/local/www/mambo/includes/core.classes.php on line 2519
[Wed Mar 11 07:39:54 2009] [error] [client 62.183.134.122] File does not exist: /usr/local/www/mambo/favicon.ico
[Wed Mar 11 07:39:57 2009] [error] [client 62.183.134.122] File does not exist: /usr/local/www/mambo/favicon.ico
^C
kasikirja@ tail -f /var/log/httpd-error.log
[Tue Mar 10 23:25:07 2009] [error] [client 88.80.7.248] File does not exist: /usr/local/www/mambo/pp
[Wed Mar 11 01:46:55 2009] [error] [client 64.71.165.195] PHP Notice: Undefined index: HTTP_USER_AGENT in /usr/local/www/mambo/includes/core.classes.php on line 2519
[Wed Mar 11 02:30:45 2009] [error] [client 213.197.183.22] File does not exist: /usr/local/www/mambo/typo3
[Wed Mar 11 02:30:45 2009] [error] [client 213.197.183.22] File does not exist: /usr/local/www/mambo/typo3
[Wed Mar 11 07:07:13 2009] [error] [client 92.62.43.77] PHP Notice: Undefined index: HTTP_USER_AGENT in /usr/local/www/mambo/includes/core.classes.php on line 2519
[Wed Mar 11 07:12:14 2009] [error] [client 92.62.43.77] PHP Notice: Undefined index: HTTP_USER_AGENT in /usr/local/www/mambo/includes/core.classes.php on line 2519
[Wed Mar 11 07:17:15 2009] [error] [client 92.62.43.77] PHP Notice: Undefined index: HTTP_USER_AGENT in /usr/local/www/mambo/includes/core.classes.php on line 2519
[Wed Mar 11 07:22:15 2009] [error] [client 92.62.43.77] PHP Notice: Undefined index: HTTP_USER_AGENT in /usr/local/www/mambo/includes/core.classes.php on line 2519
[Wed Mar 11 07:39:54 2009] [error] [client 62.183.134.122] File does not exist: /usr/local/www/mambo/favicon.ico
[Wed Mar 11 07:39:57 2009] [error] [client 62.183.134.122] File does not exist: /usr/local/www/mambo/favicon.ico
[Wed Mar 11 07:43:42 2009] [error] [client 62.183.134.122] PHP Notice: Undefined index: langfilter2 in /home/devilish/Ruttohuora.net/kysely/survey/modules.php on line 945, referer: http://devilish-develop.viishanke.fi/survey/page.php?pageid=0
[Wed Mar 11 07:43:42 2009] [error] [client 62.183.134.122] PHP Notice: Undefined index: surveynamefilter2 in /home/devilish/Ruttohuora.net/kysely/survey/modules.php on line 945, referer: http://devilish-develop.viishanke.fi/survey/page.php?pageid=0
[Wed Mar 11 07:43:42 2009] [error] [client 62.183.134.122] PHP Notice: Undefined index: surveynamefilter2 in /home/devilish/Ruttohuora.net/kysely/includes/connection.php on line 1012, referer: http://devilish-develop.viishanke.fi/survey/page.php?pageid=0
[Wed Mar 11 07:43:42 2009] [error] [client 62.183.134.122] PHP Notice: Undefined index: langfilter2 in /home/devilish/Ruttohuora.net/kysely/includes/connection.php on line 1012, referer: http://devilish-develop.viishanke.fi/survey/page.php?pageid=0
[Wed Mar 11 07:43:42 2009] [error] [client 62.183.134.122] PHP Notice: Undefined index: surveynamefilter2 in /home/devilish/Ruttohuora.net/kysely/includes/connection.php on line 1012, referer: http://devilish-develop.viishanke.fi/survey/page.php?pageid=0
[Wed Mar 11 07:43:42 2009] [error] [client 62.183.134.122] PHP Notice: Undefined index: langfilter2 in /home/devilish/Ruttohuora.net/kysely/includes/connection.php on line 1012, referer: http://devilish-develop.viishanke.fi/survey/page.php?pageid=0

```

Kuva 46. Web-palvelimelta saatavien viestien tulkintaa.

Xhtml-muotoiluvirheiden etsintään käytettiin W3Schoolsin tarjoamaa validointipalvelua. Sen avulla oli helppo tapa löytää suoranaiset syntaksivirheet ohjelman tuottamista Xhtml-sivuista.

This document was successfully checked as XHTML 1.0 Strict!	
<b>Result:</b>	Passed
<b>Source:</b>	<pre>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"&gt; &lt;html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" dir="ltr"&gt; &lt;head&gt;   &lt;link rel="stylesheet" type="text/css" href="../../includes/style.css" /&gt;   &lt;title&gt;NetSurvey&lt;/title&gt;   &lt;meta http-equiv="Content-Type" content="text/html; charset=utf-8" /&gt;   &lt;meta http-equiv="cache-control" content="no-cache" /&gt;   &lt;meta http-equiv="pragma" content="no-cache" /&gt;   &lt;script src="../../includes/prototype.js" type="text/javascript"&gt;&lt;/script&gt;   &lt;script src="../../includes/style.js" type="text/javascript"&gt;&lt;/script&gt;   &lt;link type="text/css" rel="stylesheet" href="../../includes/datechooser/javascript/dhtmlgoodies_calendar.css"&gt; &lt;/head&gt;</pre>
<b>Encoding:</b>	utf-8 (detect automatically)
<b>Doctype:</b>	XHTML 1.0 Strict (detect automatically)
<b>Root Element:</b>	html
<b>Root Namespace:</b>	<a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>

Kuva 47. W3Schoolsin tarjoama tarkastustyökalu. (W3Schools.)

Ohjelmakoodin testaaminen toteutettiin käymällä jokainen toiminto huolellisesti läpi ja miettimällä samalla, millaisia virhetoimintoja tai syötteitä käyttäjä voisi antaa lomakkeella. Tämä vaihe oli myös varsin työläs ja haastava, koska virheellisten toimintojen huomaaminen on useimmiten vaikeaa pitkän ja moneen paikkaan haarautuvan ohjelmakoodin takia.

## 4 YHTEENVETO

Ohjelman kaikki vaaditut toiminnot onnistuttiin laatimaan suunnitelman mukaisesti ja tämän lisäksi ohjelmastakin tuli yleisesti kiinnostusta herättävä, koska vastaavilla ominaisuuksilla varustettua ohjelmaa ei ole markkinoilla.

Parantamisen varaa työssä olisi ollut aikataulun suunnittelussa. Aikataulussa jouduttiin joustamaan keväällä 2008 alkaneiden työtehtävien takia, joita ei otettu huomioon alkuperäisessä aikataulutuksessa.

Projektin tulevaisuuden näkymät ovat valoisat sen herättämän kiinnostuksen takia. Tulevaisuudessa ohjelmaan lisätään mahdollisuus luoda esimerkiksi pdf- ja xls-raportteja kyselyistä, muutamia uusia kysymystyyppejä, kyselyiden ominaisuuksia ja lisää kaaviovaihtoehtoja. Muita kehitysideoita tulevaisuuden suhteen ovat kyselylomakkeiden tyylittely käyttäjän tarpeiden mukaan sekä mahdollisuus järjestää ohjelmalla suoritettavia testejä.

Projektin aikana on opittu hyödyntämään aikaisempaa tehokkaammin Php:ssä tarjolla olevia luokkia sekä funktioita. Valmiissa työssä, kuitenkin olisi voitu käyttää vieläkin pidemmälle menevää oliosuunnittelua, jossa esimerkiksi jokainen sivu olisi ollut oma olionsa ja periytetty aina yhteyspääluokasta.

## LÄHTEET

Captcha creator 2007. What is Captcha [Verkkodokumentti]. [Viitattu 17.12.2008]. Saatavissa: <http://www.captchacreator.com/>

Christoph W 2004. Storing Passwords - done right! [Verkkodokumentti]. [Viitattu 11.12.2008]. Saatavissa: <http://www.aspheute.com/english/20040105.asp>

Crypto-Gram 2004. Security Notes from All Over: Man-in-the-Middle Attack [Verkkodokumentti]. [Viitattu 1.11.2008]. Saatavissa: <http://www.schneier.com/crypto-gram-0404.html#6>

Daemen J. & Rijmen V 1999. The Rijndael Block Cipher. [Verkkodokumentti]. [Viitattu 10.11.2008]. Saatavissa: <http://www.daimi.au.dk/~ivan/rijndael.pdf>

Dhtmlgoodies 2008. A library of DHTML and Ajax scripts. [Verkkokirjasto]. [Viitattu 9.10.2008]. Saatavissa: <http://www.dhtmlgoodies.com/>

Free icons Web 2006. [Verkkodokumentti]. [Viitattu 1.11.2008]. Saatavissa: <http://www.freeiconsweb.com/>

Gervasio A. 2006. The Basics of Serializing Objects in PHP. [Verkkodokumentti]. [Viitattu 3.1.2009]. Saatavissa: <http://www.devshed.com/c/a/PHP/The-Basics-of-Serializing-Objects-in-PHP/>

InnoDB 2008. InnoDB. [Verkkodokumentti]. [Viitattu 21.1.2009]. Saatavissa: <http://www.innodb.com/>

Kekkonen A 2009. Mysql materiaali. [Verkkodokumentti]. [Viitattu 10.3.2009]. Saatavissa: <http://www.ratol.fi/opensource/mysql/avaimet.htm>

Leponiemi J 2002. HTTP ja PHP, palvelupyyntöjen käsittely. [Verkkodokumentti]. [Viitattu 9.10.2008]. Saatavissa: <http://www11.uta.fi/~jl/php/http.php>

Mysql 2009. Mysql 5.0 Reference Manual. [Verkkodokumentti]. [Viitattu 1.11.2008]. Saatavissa: <http://dev.mysql.com/doc/>

PChart 2008. SQL Documentation. [Verkkodokumentti]. [Viitattu 9.11.2008]. Saatavissa: <http://pchart.sourceforge.net/documentation.php>

Php 2009. `mysql_real_escape_string` — Escapes special characters in a string for use in a SQL statement. [Verkkodokumentti]. [Viitattu 1.1.2009]. Saatavissa: [http://fi.php.net/mysql\\_real\\_escape\\_string](http://fi.php.net/mysql_real_escape_string)

Prototypejs 2009. Prototype API Documentation [Verkkodokumentti]. [Viitattu 1.1.2009]. Saatavissa: <http://www.prototypejs.org/>

Reverse MD5 hash lookup 2009. [Verkkodokumentti]. MD5-sanakirja. [Viitattu 3.2.2009]. Saatavissa: <http://tools.benramsey.com/md5/>

Rivest R 1992. The MD5 Message-Digest Algorithm. [Verkkodokumentti]. [Viitattu 1.12.2008]. Saatavissa: <http://tools.ietf.org/html/rfc1321>

Tizag 2008. SQL Tutorial. [Verkkodokumentti]. [Viitattu 1.11.2008]. Saatavissa: <http://www.tizag.com/mysqlTutorial/mysql-php-sql-injection.php>

Tizag 2008. Php/mysql Tutorial. [Verkkodokumentti]. [Viitattu 1.11.2008]. Saatavissa: <http://www.tizag.com/mysqlTutorial/>

Trémeaux J-M 2007. Reference API. [Verkkodokumentti]. [Viitattu 25.9.2008]. Saatavissa: <http://naku.dohcrew.com/libchart/pages/documentation/>

W3Schools 2009. XHTML Tutorial. [Verkkodokumentti]. [Viitattu 1.1.2009]. Saatavissa: [http://www.w3schools.com/xhtml/xhtml\\_intro.asp](http://www.w3schools.com/xhtml/xhtml_intro.asp)

# LIITTEET

## Tietokannan Sql-lausekkeet

```
CREATE TABLE IF NOT EXISTS `netsurvey_admin` (  
  `username` varchar(10) character set utf8 collate utf8_unicode_ci  
  NOT NULL,  
  `password` varchar(255) character set utf8 collate utf8_unicode_ci  
  NOT NULL,  
  `specialrights` enum('admin','none') character set utf8 collate  
  utf8_unicode_ci NOT NULL,  
  `email` varchar(65) character set utf8 collate utf8_unicode_ci NOT  
  NULL,  
  `firstname` varchar(60) character set utf8 collate utf8_unicode_ci  
  default NULL,  
  `lastname` varchar(60) character set utf8 collate utf8_unicode_ci  
  default NULL,  
  `activity` enum('yes','no') character set utf8 collate utf8_uni-  
  code_ci NOT NULL,  
  `streetaddress` varchar(75) character set utf8 collate utf8_uni-  
  code_ci NOT NULL,  
  `zipcode` varchar(10) character set utf8 collate utf8_unicode_ci  
  default NULL,  
  `city` varchar(65) character set utf8 collate utf8_unicode_ci de-  
  fault NULL,  
  `telephone` varchar(45) character set utf8 collate utf8_unicode_ci  
  default NULL,  
  `country` varchar(45) default NULL,  
  `state` varchar(45) character set utf8 collate utf8_unicode_ci de-  
  fault NULL,  
  `sessionid` varchar(45) character set utf8 collate utf8_unicode_ci  
  default NULL,  
  PRIMARY KEY (`username`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE IF NOT EXISTS `netsurvey_answerer` (  
  `answererid` bigint(20) unsigned NOT NULL auto_increment,  
  `firstname` varchar(45) collate utf8_unicode_ci NOT NULL,  
  `lastname` varchar(45) collate utf8_unicode_ci NOT NULL,  
  `company` varchar(50) collate utf8_unicode_ci NOT NULL,  
  `email` varchar(60) collate utf8_unicode_ci NOT NULL,  
  `active` enum('yes','no') collate utf8_unicode_ci NOT NULL,  
  `password` varchar(255) collate utf8_unicode_ci NOT NULL,  
  `activationid` varchar(255) collate utf8_unicode_ci default NULL,  
  `streetaddress` varchar(60) collate utf8_unicode_ci NOT NULL,  
  `zipcode` varchar(12) collate utf8_unicode_ci NOT NULL,  
  `city` varchar(50) collate utf8_unicode_ci NOT NULL,  
  `state` varchar(40) collate utf8_unicode_ci NOT NULL,  
  `country` varchar(40) collate utf8_unicode_ci NOT NULL,  
  `telephone` varchar(15) collate utf8_unicode_ci NOT NULL,  
  `department` varchar(45) collate utf8_unicode_ci NOT NULL,  
  `sessionid` varchar(45) collate utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`answererid`),  
  UNIQUE KEY `email` (`email`)
```



```
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=1 ;
```

```
CREATE TABLE IF NOT EXISTS `netsurvey_answerergroup` (
  `groupid` bigint(20) unsigned NOT NULL auto_increment,
  `groupname` varchar(32) collate utf8_unicode_ci NOT NULL COMMENT
'Name of group',
  PRIMARY KEY (`groupid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=4 ;
```

```
CREATE TABLE IF NOT EXISTS `netsurvey_answers` (
  `answerid` bigint(20) unsigned NOT NULL auto_increment,
  `questionid` bigint(20) unsigned NOT NULL,
  `ip` varchar(128) collate utf8_unicode_ci NOT NULL,
  `answer` longtext collate utf8_unicode_ci NOT NULL,
  `activated` enum('yes','no') collate utf8_unicode_ci default NULL,
  `date` date NOT NULL,
  `answererid` bigint(20) unsigned default NULL,
  `ordernumber` mediumint(8) unsigned default NULL COMMENT '''This
will be used in listings, when privacy of user must be
guaranteed''',
  `activationcode` varchar(250) collate utf8_unicode_ci NOT NULL
COMMENT 'activationcode',
  `email` varchar(50) collate utf8_unicode_ci default NULL COMMENT
'anonymous-answerers-emailaddress',
  PRIMARY KEY (`answerid`),
  KEY `questionid` (`questionid`),
  KEY `answererid` (`answererid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=1;
```

```
CREATE TABLE IF NOT EXISTS `netsurvey_questions` (
  `questionid` bigint(20) unsigned NOT NULL auto_increment,
  `surveyid` bigint(20) unsigned NOT NULL,
  `ordernumber` mediumint(8) unsigned default NULL,
  `question` longtext collate utf8_unicode_ci NOT NULL,
  `help` longtext collate utf8_unicode_ci NOT NULL,
  `type`
enum('zerotofive','yesorno','yesnonone','select','text','option')
collate utf8_unicode_ci NOT NULL,
  `selectbox` longtext collate utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`questionid`),
  KEY `surveyid` (`surveyid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=1 ;
```

```
CREATE TABLE IF NOT EXISTS `netsurvey_rights` (
  `username` varchar(10) collate utf8_unicode_ci NOT NULL,
  `surveyid` bigint(20) unsigned NOT NULL,
  `rightnumber` bigint(20) unsigned NOT NULL auto_increment,
  `userright` enum('read','modify') collate utf8_unicode_ci NOT
NULL,
  PRIMARY KEY (`rightnumber`),
  UNIQUE KEY `unique` (`username`,`surveyid`),
  KEY `survey` (`surveyid`),
  KEY `user` (`username`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=1 ;
```

```
CREATE TABLE IF NOT EXISTS `netsurvey_surveys` (
  `surveyid` bigint(20) unsigned NOT NULL auto_increment,
  `language` varchar(40) character set utf8 collate utf8_unicode_ci
  NOT NULL,
  `description` longtext character set utf8 collate utf8_unicode_ci
  NOT NULL,
  `surveyname` varchar(255) character set utf8 collate
  utf8_unicode_ci NOT NULL,
  `password` varchar(255) character set utf8 collate utf8_unicode_ci
  default NULL,
  `starts` datetime default NULL,
  `ends` datetime default NULL,
  `anonymousanswering` enum('yes','no') character set utf8 collate
  utf8_unicode_ci NOT NULL default 'no',
  `active` enum('yes','no') character set utf8 collate
  utf8_unicode_ci NOT NULL default 'yes',
  `emailconfirmation` enum('yes','no') NOT NULL default 'yes',
  `visible` enum('yes','no') NOT NULL default 'no',
  `groupid` bigint(20) unsigned default NULL,
  PRIMARY KEY (`surveyid`),
  KEY `groups` (`groupid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

```
CREATE TABLE IF NOT EXISTS `netsurvey_usermembership` (
  `membershipid` bigint(20) unsigned NOT NULL auto_increment,
  `groupid` bigint(20) unsigned NOT NULL,
  `answererid` bigint(20) unsigned NOT NULL,
  PRIMARY KEY (`membershipid`),
  KEY `groupid` (`groupid`),
  KEY `answererid` (`answererid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=1 ;
```

```
ALTER TABLE `netsurvey_answers`
  ADD CONSTRAINT `netsurvey_answers_ibfk_1` FOREIGN KEY (`ques-
  tionid`) REFERENCES `netsurvey_questions` (`questionid`) ON DELETE
  CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `netsurvey_answers_ibfk_2` FOREIGN KEY (`answer-
  erid`) REFERENCES `netsurvey_answerer` (`answererid`) ON DELETE SET
  NULL ON UPDATE CASCADE;
ALTER TABLE `netsurvey_questions`
  ADD CONSTRAINT `netsurvey_questions_ibfk_1` FOREIGN KEY (`survey-
  id`) REFERENCES `netsurvey_surveys` (`surveyid`) ON DELETE CASCADE
  ON UPDATE CASCADE;
ALTER TABLE `netsurvey_rights`
  ADD CONSTRAINT `netsurvey_rights_ibfk_1` FOREIGN KEY (`username`)
  REFERENCES `netsurvey_admin` (`username`) ON DELETE CASCADE ON UP-
  DATE CASCADE,
  ADD CONSTRAINT `netsurvey_rights_ibfk_2` FOREIGN KEY (`surveyid`)
  REFERENCES `netsurvey_surveys` (`surveyid`) ON DELETE CASCADE ON UP-
  DATE CASCADE;
ALTER TABLE `netsurvey_surveys`
  ADD CONSTRAINT `netsurvey_surveys_ibfk_1` FOREIGN KEY (`groupid`)
  REFERENCES `netsurvey_answerergroup` (`groupid`) ON DELETE SET NULL
  ON UPDATE CASCADE;
ALTER TABLE `netsurvey_usermembership`
```

```
ADD CONSTRAINT `netsurvey_usermembership_ibfk_2` FOREIGN KEY (`answererid`) REFERENCES `netsurvey_answerer` (`answererid`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `netsurvey_usermembership_ibfk_3` FOREIGN KEY (`groupid`) REFERENCES `netsurvey_answerergroup` (`groupid`) ON DELETE CASCADE ON UPDATE CASCADE;
```